applied sciences

MDPI

Article

Combining Machine Learning and Social Network Analysis to Reveal the Organizational Structures

Mateusz Nurek and Radosław Michalski *

Department of Computational Intelligence, Faculty of Computer Science and Management, Wrocław University of Science and Technology, 50-370 Wrocław, Poland; mateusz.nurek@pwr.edu.pl

* Correspondence: radoslaw.michalski@pwr.edu.pl

Received: 7 January 2020; Accepted: 21 February 2020; Published: 2 March 2020

Abstract: Formation of a hierarchy within an organization is a natural way of assigning the duties, delegating responsibilities and optimizing the flow of information. Only for the smallest companies the lack of the hierarchy, that is, a flat one, is possible. Yet, if they grow, the introduction of a hierarchy is inevitable. Most often, its existence results in different nature of the tasks and duties of its members located at various organizational levels or in distant parts of it. On the other hand, employees often send dozens of emails each day, and by doing so, and also by being engaged in other activities, they naturally form an informal social network where nodes are individuals and edges are the actions linking them. At first, such a social network seems distinct from the organizational one. However, the analysis of this network may lead to reproducing the organizational hierarchy of companies. This is due to the fact that that people holding a similar position in the hierarchy possibly share also a similar way of behaving and communicating attributed to their role. The key concept of this work is to evaluate how well social network measures when combined with other features gained from the feature engineering align with the classification of the members of organizational social network. As a technique for answering this research question, machine learning apparatus was employed. Here, for the classification task, Decision Trees, Random Forest, Neural Networks and Support Vector Machines have been evaluated, as well as a collective classification algorithm, which is also proposed in this paper. The used approach allowed to compare how traditional methods of machine learning classification, while supported by social network analysis, performed in comparison to a typical graph algorithm. The results demonstrate that the social network built using the metadata on communication highly exposes the organizational structure.

Keywords: social network analysis; classification; organizational hierarchy; machine learning

1. Introduction

People around the world send hundreds of emails to exchange information within organizations. As an implicit result of that, each of these interactions forms a link in a social network. This network can be a valuable source of knowledge about human behaviors and what is more, conducting the analysis can reveal groups of employees with similar communication patterns. These groups usually coincide with different levels of the organization's hierarchy and additionally, employees who work in the same position generally have a comparable scope of duties. It is common for organizations to observe some hierarchy because, formally, an organized structure helps with better management of employees and gaining an advantage within the market. Therefore, the analysis of the network created from a set of emails could retrieve valuable data about inner corporation processes and recreate an organizational structure. An interesting and promising idea seems to be the combination of network measures and additional features extracted from messages for classification tasks. Social network analysis (SNA) has the potential to boost machine learning algorithms in a field of organization structure detection, thus capturing relations between data seems to be very important for this kind of dataset.

The reverse engineering of the corporate structure of an organization can be perceived two-way. On the one hand, if successful, it could reveal company structure by having only meta-data and this imposes a risk in the case when the structure is intentionally kept secret, for example, for keeping the competitive edge or protecting the employees from takeovers by other companies. On the other hand, this could lead to reconstructing the structure of malicious organizations when only partial information about them is available.

In the literature, there are several works describing the detection of organizational structures. However, most of them use the Enron dataset [1] or focus rather on a network approach and omit standard supervised classification algorithms. It should be noted that each organization is managed in a slightly different way which means that communication patterns could differ within each of them. These differences imply that some solutions may give better or worse results depending on the network's specificity; it is important that studies on an organization hierarchy should not be limited to only one dataset.

The authors of Reference [2,3] introduced a concept of matching a formal organizational structure and a social network created from email communications. Experiments were carried out on messages coming from a manufacturing company located in Poland as well as the well-known Enron dataset. The research results showed that in both cases, some network metrics were able to reveal organizational hierarchy better than others. This work also touched on the problem that a formal structure sometimes may significantly differ and will not converge with the daily reality.

The idea of combining network metrics and other features extracted from email dataset to reveal corporate hierarchy is introduced in Reference [4]. The authors presented their own metric named "social score" which defines the importance of each employee in the network. This metric is defined as a weighted average of all features and is used in a grouping algorithm. The grouping method is a simple straight scale level division algorithm which assigns employees to defined intervals by the social score.

The study on the usage of network measures as input features for classification algorithms was presented in Reference [5]. The basic concept of this work focused on retrieving company hierarchy based on the network created from social media accounts of the employees. The authors presented that centrality measures and clustering coefficients in combination with other features extracted from social media can detect leaders in a corporate structure. However, this research used individual features of a person like a gender, hometown or number of friends instead of features gained from job activities and interactions among employees. Other articles describing the combination of SNA and standard classification methods are References [6,7]. They both work using the Enron dataset and features based on the number of sent/received messages. In Reference [8] the usage of some network metrics as input for classification and clustering algorithms has been described. Furthermore, the results were compared to a novel measure called Human Rank (improvement of Page Rank). However, the use of classification based on social network features is not limited only to the corporate environment. For instance, following the ideas of studying the social networks of criminalists [9], the authors of Reference [10] used features of a social network of co-arrestees for predicting the possibility of future violent crimes. A similar concept was also used in Reference [11] for analyzing co-offending networks. In that work, a co-offence prediction algorithm using supervised learning has been developed. Yet, the classification in social networks based on communication or behaviour in social media, can relate to completely different areas, such as poverty detection [12], personality traits discovery [13] or occupation [14]. All that is possible because our digital traces do differ depending on our role or status.

In the area of the problem being solved also many solutions concentrated mainly on classification from a graph perspective. For instance, identification of key players of social network based on entropy [15], applying graphical models [1] or factor graph models [16].

The following work focuses on the organizational structure detection based on nine-months of e-mail communication between employees of a manufacturing company located in Poland as well as the Enron dataset. The research used Decision Tree, Random Forest, Neural Network and Support Vector Machine (SVM) algorithm for classification, moreover influence of minimum employee activity was examined. The obtained results were compared with the simple graph algorithm of collective classification also proposed in this paper. The weakness of this approach is the fact that an independent and identical distribution (IID) condition is difficult to meet due to network measures which were calculated once before splitting data on training and test set. In social network analysis, full satisfaction of the IID condition is hard to achieve because if we had built independent networks for training and test data, we would get totally different network measures and the importance of each node could be biased. However, network measures could be valuable features for machine learning algorithms in sight of capturing connections between data. The results showed that the combination of classification algorithm and social network analysis can reveal organizational structures, however, small changes in the network can change the efficiency of the algorithms. Furthermore, a graph approach, such as collective classification, is able to classify well even with limited knowledge about node labels.

2. Materials and Methods

In this section, after introductory sections to supervised learning and social network analysis, a proposed solution is described in detail, as well as the used datasets. The presented solution is created with Python language, as well as NetworkX library for a social network creation and Scikit-learn for all machine learning tasks.

2.1. Supervised Learning

Machine learning can be considered an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed [17]. The set of tools derived from the field of statistics enables the possibility to perform multiple tasks far exceeding human capabilities or simple algorithms in variety of disciplines, ranging from text analysis, computer vision, medicine and others. In machine learning, classification is a supervised learning approach in which the algorithm learns from the data input given to it and then uses this learning to classify new observation. Here, by supervised we mean providing the algorithm instances of objects that have been categorized as belonging to certain class and requiring it to develop a method to adequately classify other objects without known class. In order to fulfill this goal, numerous algorithms have been developed and tuned over last decades, such as logistic regression [18], naive Bayes classifier [19], nearest neighbor [20], Support Vector Machines [21], decision trees [22], random forests [23] or neural networks [24]. Each of these methods takes a different perspective to the task. Regarding the methods used in this work, decision trees build a tree consisting of the tests of features: each branch represents the outcome of the test, and each leaf node represents a class label. Random forest extends the concept of decision trees by building a multitude of them and outputs the class that is the mode of the classes of the individual trees. What one can say about these two methods is that the rules of classification are transparent and highly interpretable. Regarding two other methods evaluated in this work, Support Vector Machine constructs a hyperplane or a set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. Neural networks try to mimic human brain in terms of how information is being passed and analysed. Here, a neural network is based on a collection of connected units or nodes called neurons. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. Neurons are grouped in layers and a signal passing the layers is being converted by neurons into the one that at the final (output layer) will be decided upon the class membership. This creates an architecture of neural network. Contrary to decision trees and random forests, Support Vector Machines and neural networks are not that easily interpretable in terms of the importance of features [25].

What links all of these methods is that they usually require that the samples are required to follow IID principle, namely they have to be independent and identically distributed. Unfortunately, this is not always the case, especially when we consider any network-related data. For instance in social networks, people tend to cluster in groups of similar interests [26] or change their opinion based on others' opinion [27]. In this case it is hard to consider the samples as IID, so another set of approaches has been developed: collective classification that tries to make the use of the networked structure of the data [28].

Another problem in classification is that rarely the instances are equally distributed over all classes to be classified. This problem is referred to as imbalanced data and there are multiple techniques that allow to tackle it, mainly belonging to one of two groups: under-sampling and over-sampling [29,30]. In under-sampling, the dominant groups are being reduced to be equally represented as previously under-represented classes. Contrary, over-sampling generates the synthetic instances that belong to under-represented class leading to more balanced data. One of the most prominent over-sampling techniques is SMOTE that bases on nearest neighbors judged by Euclidean Distance between data points in feature space and perform vector operations to generate new data points [31,32]. Using this technique provides more representative and less biased samples compared to random over-sampling.

In Section 2.6 one can find information on which classification algorithms have been used in this work and Section 2.7 contains more information on how we used collective classification for discovering the organizational structure from social network.

2.2. Social Network Analysis

The field of social network analysis can be understood as a set of techniques deriving knowledge about human relationships based on the relations they form—usually by being members of social networks of different kinds. These networks can relate to family, friends, companies or organizations they are employees members of, or social media they participate in. More formally, social network consists of a finite set or sets of actors and the relation or relations defined on them [33]. To help understanding this definition of a social network, some other concepts that are fundamental in this case should be explained. An actor is a discrete individual, corporate or collective social unit [33]. This can be a person in a group of people, a department within a company or a nation in the world system. Actors are linked to each other by social ties and these relations are the core of the social network approach. Social networks are presented using graph structures, where nodes are actors and edges are connections between them. Hence, all graph theory methods and measured can be applied. A graph may be undirected, which means that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another. A graph is being defined usually as an ordered pair $G := (V, E)$, where V are vertices or nodes and E are edges. On top of that multiple methods are being used in order to derive knowledge about network members or network itself, such as centrality measures [34], community detection [35], modelling evolution [36], or detection of influential nodes [37]. As described in Section 1, social network analysis also became present in organizations where it is often being referred to as organizational network analysis [38].

2.3. Datasets

In this work we evaluated two datasets containing both: metadata on communication and organizational structure in companies. This allowed to use the features extracted from the social network built using the communication data as features for classifiers. These classifiers have been then distinguishing the level of an employee in a corporate hierarchy. Detailed description of the datasets is presented below.

2.3.1. Manufacturing Company

The analyzed dataset contains a nine-month exchange of messages among clerical employees of a manufacturing company located in Poland [3]. The dataset consists of two files—the first contains the company hierarchy, the second stores the communication history. The analyzed company contains the three level hierarchy: the first management level, the second management level and regular employees. The file with emails consists of senders and recipients, as well as the date and time of sent messages. Moreover, emails from former employees and technician accounts are also included in this file. Due to the lack of data about supervisors, former employees and technical accounts were removed from the further research. The final organizational structure to be analyzed is shown in Figure 1 and in Table 1. It is important to note that the dataset does not contain any correspondence with anyone outside of the company, moreover, the company structure has been consistently stable within the period of time being considered and has not undergone any changes.

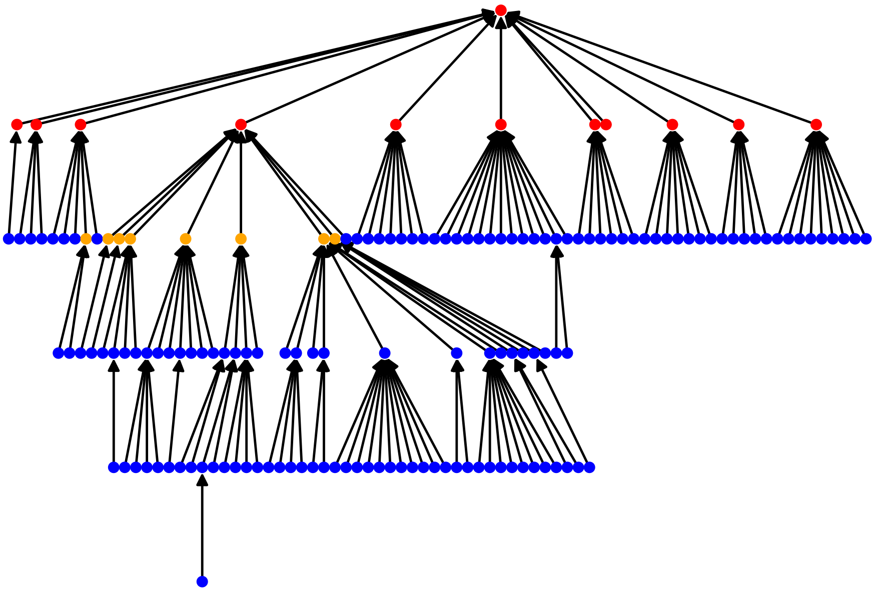


Figure 1. Organizational hierarchy after removal of former employees and technical accounts. Red nodes—first level management. Orange nodes—second level management. Blue nodes—regular employees.

Table 1. Organizational structure after removal of former employees and technical accounts.

Hierarchy Level	Number
The first management level	12
The second management level	8
Regular employees	134

2.3.2. Enron

The second analyzed dataset comes from the Enron company [1]. Enron was a large American energy establishment founded in 1985 subsequently became famous at the end of 2001 due to financial fraud. During the investigation, the dataset has been made public, however the organizational hierarchy has never been officially confirmed. Despite of this limitation, the Enron email corpus has become the subject of many studies, which allowed to partially reconstruct the company's structure. The authors of this paper decided to use processed version of this dataset which already include positions assigned to the employees. There is a seven-level hierarchy in this data set, however, to reduce the complexity of this structure the authors proposed more generic three-level hierarchy showed in Table 2, the same as in the manufacturing company dataset. The applied approach allowed for a better distinction of managerial and executive positions from regular employees. The analyzed period contains messages from over 3 years and due to limited knowledge about inner company processes the authors assumed that the organizational structure was stable during it.

Table 2. Enron hierarchy.

Flattened	Original	Number
The first management level	CEO President Vice President	40
The second management level	Director Managing Director Manager	37
Regular employee	Employee In House Lawyer Trader	53

Both datasets are available for evaluation or further research, see Supplementary Materials.

2.4. Network

The network was built using the email exchanges of its members, where the nodes were employees and the edges were the messages. It was decided to use a directed graph defined as follows: Social network is a tuple $SN = (V, E)$, where $V = \{v_1, \dots, v_n\}, n \in \mathbb{N}_+$ is the set of vertices and $E = \{e_1, \dots, e_{k^e}\}, k^e \in \mathbb{N}_+$ is the set of edges between them. Each vertex $v_i \in V$ represents an individual v_i^e and each edge e_{ij} corresponds to the directed social relationship from v_i to v_j, such that $E = \{(v_i, v_j, w_{ij}) : v_i \in V, v_j \in V, v_i = v_i^e, v_j = v_j^e$ and $w_{ij} \in [0, 1]\}$. The edge weights defined according to the following formula:

$$w_{ij} = \frac{\sum e_{ij}}{\sum e_i}, \tag{1}$$

where $\sum e_{ij}$ is the sum of messages sent from node i to node j and $\sum e_i$ is the total sum of messages sent from node i. All self loops were removed.

In Figure 2 the weighted directed network built using e-mail communication in the manufacturing company is depicted. What can be noticed is that the position of the first level and second level management is not always central in this network. As a result of that, using centrality measures would not be enough for detecting positions in organizational hierarchy. The reasons why there is no direct correlation between position in the social network and the organizational networks can be of many kind, for example, management positions do not require intense communication, using different forms of communication or having supporting personnel to communicate on behalf.

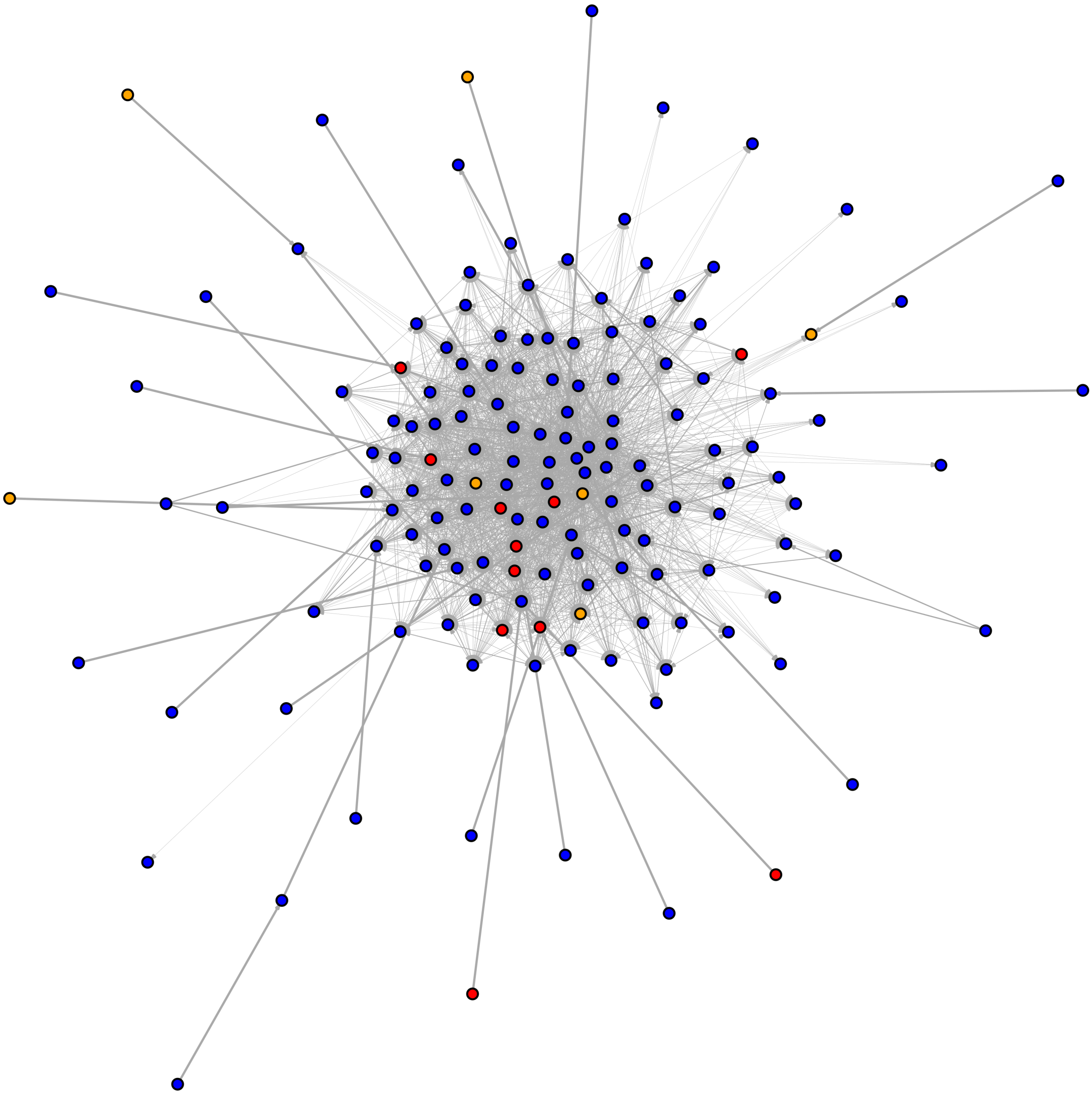


Figure 2. Weighted and directed social network in a manufacturing company built upon e-mail communication. The colouring scheme is the same as for Figure 1: red nodes—first level management, orange nodes—second level management, blue nodes—regular employees. The algorithm used for visualisation is a force-directed large graph layout [39] with the root node with the highest betweenness.

2.5. Features

In the created social network, the centrality measures presented below have been calculated as input features for classification algorithms. These measures are also briefly described in Table 3.

- indegree centrality:

$$C_{IN}(v_i) = |e_{ji} \in E|, j \neq i, \tag{1}$$

where e_{ji} is the edge going from every node v_j to evaluated node v_i.

- outdegree centrality:

$$C_{OUT}(v_i) = |e_{ij} \in E|, i \neq j, \tag{2}$$

where e_{ij} is the edge going from evaluated node v_i to every other node v_j in the network.

- betweenness centrality:
$$C_B(v_i) \;=\; \sum_{v_s \neq v_i \neq v_d} \frac{\sigma_{v_s v_d}(v_i)}{\sigma_{v_s v_d}}, \tag{3}$$
where $\sigma_{v_s v_d}(v_i)$ is the number of shortest paths between nodes v_s and v_d passing through node v_i and $\sigma_{v_s v_d}$ is the number of all shortest paths between v_s and v_d.

- closeness centrality:
$$C_C(v_i) \;=\; \frac{N}{\sum_{v_y} d(v_y, v_i)}, \tag{4}$$
where N is the number of vertices in the network and $d(v_y, v_i)$ is a distance between vertices v_y and v_i.

- eigenvector centrality:
$$C_E(v_i) \;=\; \frac{1}{\lambda} \sum_k a_{v_k, v_i} C_E(v_k), \tag{5}$$
where $A \;=\; (a_{i,j})$ is the adjacency matrix of a graph and $\lambda \neq 0$ is a constant.

- page rank:
$$C_{PR}(v_i) \;=\; \alpha \sum_k \frac{a_{v_k, v_i}}{d_k} C_{PR}(v_k) + \beta, \tag{6}$$
where α and β are constants and d_k is the out-degree of node v_k if such degree is positive, or $d_k \;=\; 1$ if the out-degree of node v_k is null. Again, $A \;=\; (a_{i,j})$ is the adjacency matrix of a graph.

- hub centrality:
$$C_{HUB}(v_i) \;=\; \beta \sum_k a_{v_i, v_k} C_{AUT}(v_k), \tag{7}$$
where $A \;=\; (a_{i,j})$ is the adjacency matrix of a graph and $C_{AUT}(v_k)$ is the authority centrality of a node (see Equation (8)), β is a constant.

- authority centrality:
$$C_{AUT}(v_i) \;=\; \alpha \sum_k a_{v_k, v_i} C_{HUB}(v_k), \tag{8}$$
where $A \;=\; (a_{i,j})$ is the adjacency matrix of a graph and $C_{HUB}(v_k)$ is the hub centrality of a node (see Equation (7)), α is a constant.

Moreover, a local clustering coefficient was calculated for each node, which allows capturing density of connections between neighbors, as well as two additional features related to cliques:

$$C_{CC}(v_i) \;=\; \frac{2m_{v_i}}{k_i(k_i - 1)}, \tag{9}$$

where m_{v_i} is the number of pairs of neighbors of a node v_i that are connected. In the formula it is linked with the the number of possible pairs of neighbors of node v_i, which is $\frac{k_{v_i}(k_{v_i}-1)}{2}$, where k_{v_i} is the degree of a node v_i.

A clique is defined as a fully connected subgraph which means that each node has directed links to all other nodes in the clique. The first feature is the total numbers of cliques in which an employee is assigned, furthermore the second is the size of the biggest clique for the specific node. Reference [40] contains more details on all the measures introduced above.

The next features were based on neighborhood variability, which is determined in three ways: sent neighborhood variability, received neighborhood variability and general neighborhood variability.

Overall, neighborhood variability is defined as the difference between sets of neighbors which the specific node communicates in the previous and the next month. Sent neighborhood variability considers a set of neighbors to which the given node was sending messages. Received neighborhood variability looks at a set of neighbors from which the given node had been receiving messages. General neighborhood variability uses a set of neighbors with which the node communicates, without distinguishing between sending and receiving messages. The Jaccard coefficient was used for calculating the difference between sets, so the coefficient takes values between 0 and 1 where 0 means totally different sets and 1 means identical sets. The Jaccard coefficient was calculated for each pair of alternating months. Moreover, if the employee had not been active in a directly following month, the nearest next month would be considered. For example: the employee was active in January, but not in February and again was active in March; therefore, coefficient would be calculated between sets of neighbors in January and March. Furthermore, a neighborhood variability was calculated as an average Jaccard coefficient for each node based on previous partial coefficients. Formally, sent variability measure can be defined as following:

$$VAR_{SNT}(v_i) = \frac{|N_{snt_{v_i,m-1}} \cap N_{snt_{v_i,m}}|}{|N_{snt_{v_i,m-1}} \cup N_{snt_{v_i,m}}|}, \tag{10}$$

where $N_{snt_{v_i,m-1}}$ is the set of neighbours that a certain node v_i sent messages to in the month $m-1$ and $N_{snt_{v_i,m}}$ is the is the set of neighbours that a certain node v_i sent messages to in month m or, if no messages have been sent in m, then $m+1, m+2, \ldots, m_{max}$ are considered. Similarly, received and general neighbourhood variability measures can be defined by substituting the sets of neighbours to the neighbours that either sent messages to node v_i (received variability) or the set of neighbours the contact occurred with in any direction and involved v_i (general variability).

Furthermore, features such as the number of weekends worked and the amount of overtime taken were taken into account. For overtime, work between 4:00 PM and 6:00 AM were considered. It should be mentioned that overtime was only calculated for the manufacturing company dataset. Due to the limited knowledge about the Enron dataset, it was impossible to know whether different timezones should be considered because the dates were given in the POSIX format and Enron had branches located in different timezones.

As a summary, all used features used for classification are presented in Table 3.

Table 3. Features.

Feature Name	Defined in	Brief Description
indegree centrality	Equation (1)	a number of incoming links to a given node
outdegree centrality	Equation (2)	a number of outgoing links from a given node
betweenness centrality	Equation (3)	the frequency of a node appearing in shortest paths in the network
closeness centrality	Equation (4)	the length of the shortest paths between the node and all other nodes in the graph
eigenvector centrality	Equation (5)	a relative measure of importance dependent on the importance of neighbouring nodes in the network
page rank centrality	Equation (6)	relative measure of importance also based on eigenvectors of an adjacency matrix, more tunable
hubs centrality	Equation (7)	indication of position in relevance to important nodes—authorities

Table 3. *Cont.*

Feature Name	Defined in	Brief Description
authorities centrality	Equation (8)	importance of node based on being referred to by hubs
clustering coefficient	Equation (9)	degree to which nodes in a graph tend to cluster together
the total numbers of cliques	Section 2.5	total numbers of cliques in which an employee is assigned
the biggest clique	Section 2.5	size of the biggest clique for the specific node
sent neighborhood variability	Equation (10)	difference between sets of neighbours a node sends emails to in consecutive months
received neighborhood variability	Equation (10)	difference between sets of neighbours a node receives emails from in consecutive months
general neighborhood variability	Equation (10)	difference between sets of neighbours a node communicates with in consecutive months
overtime	Section 2.5	a number of days an employee worked overtime (only for manufacturing company)
the number of weekends worked	Section 2.5	how many times an employee worked over weekends

2.6. Classification

The classification task was carried out using the Decision Tree [22], Random Forest, Neural Network (multi-layer perceptron) with L-BFGS solver and SVM algorithm with the polynomial kernel for different set up of following parameters of the experiment: number of recognized employee groups, minimum number of active months as well as the percentage of used features.

The first parameter refers to the previously mentioned three-level hierarchy of employees, which can also be flattened to only two levels—management level and regular employees. The experiment was run with two values of this parameter to see how the performance of the algorithms vary with recognizing two and three groups of employees.

The meaning of the second parameter is checked to see how the activity of a person may have influence on the result of the classification and therefore was examined to see if higher minimum months of employee activity correspond with better results. There is an assumption that some patterns of behavior required more time to be revealed, so the classification was run five times starting with one month minimum activity and ending with 5 months minimum activity. For each value, the network had to be recreated and features calculated again as some nodes were eliminated from the network.

The third parameter examines the impact of the elimination of the most significant features. For this parameter, the experiment was carried out nine times, starting from all features to only ten percent of features with a continual decrease of ten percent. The importance of features for Decision Tree and Random Forest algorithms was determined based on Gini importance parameter from previously learned model. The Neural Network and SVM algorithm are not so easily interpretable and importance of the features cannot be obtained from the outcome of the model. In this case, the importance of the features must be determined before learning a model. For this purpose, the univariate feature selection method based on the chi-squared test was used.

In the analyzed manufacturing company dataset, there was a problem with the unbalanced size of classes, which is common for a company structure where the group of regular employees is the most numerous and the management level has fewer members. However, the Enron dataset is much more balanced in each group, which may indicate a different management model in this company. To handle this problem the technique of oversampling was used to solve it, therefore to match the size of all minor classes to the size of the majority class of regular employees, SMOTE algorithm was used. To prevent data leakage, oversampling was performed only on a training set.

In general, for each combination of the above parameters, a model was trained with the usage of the grid search algorithm with 5-fold cross validation, so all the possible combinations from the range of given values were tested, and the best one with respect to the f-score macro average was returned. The hyperparameters search space is shown in Table A21 as well as the best ones for each model in Tables A22–A25.

2.7. Collective Classification

Collective classification is a different way of revealing company hierarchy from a graph perspective. This approach uses the connection between nodes to propagate labels within the whole network. Loopy belief propagation is an example of collective classification described in detail in References [28,41]. Therefore, in this paper, a simplified version of this algorithm is introduced to compare with standard classification algorithms.

The proposed collective classification method is presented as Algorithm 1. The first step of this algorithm is choosing a utility score and sorting all nodes according to it (line 1). The utility score can be one of the calculated features from the previous section. The next step is to reveal the labels for the given percentage of nodes of each class $l_i \in L$ with the highest utility score (line 2). These nodes are marked as known (their labels are constant) and labeled V^L , whereas the other nodes are treated as unkown V^{UK} and unlabeled. Furthermore, the propagation of labels begins in a loop until the stop condition is met or the number of iterations exceeds the given maximum number of iterations. In one iteration, each labeled node sends a message to all of its neighbors by treating edges as undirected (line 5); moreover, all received labels in a given iteration are saved for each node v_i in a counter c_{v_i} (line 6). The labels update begins after all nodes sent a message to their neighbors, so the sending order does not affect the result. If the node v_i has received one label more often than others (line 13), this label will be assigned to it (line 14) and the node will be additionally treated as labeled $v_i \in V^L$ (line 15), otherwise for this node counter u_{v_i} will be increased (line 18). If u_{v_i} exceeds the maximum value (line 20), it will be reset (line 26) and the node will be assigned the label with the highest position in the company hierarchy among the labels with the highest count (lines 22 to 24). At the end of iteration the stop conditions is always checked and it is determined as a difference between sets of previous and current labels, therefore if the Jaccard coefficient is bigger than the given minimum Jaccard value and all nodes have assigned label then the algorithm will end (line 29). Additionally, in the case of unbalanced classes, the algorithm allows defining a *threshold*. During the phase of counting how many times each label was received by the node, the result for the majority class will be divided by this threshold to prevent domination of this class. (line 11)

The collective classification algorithm was run with three parameters: number of recognized employee groups, minimum number of active months, percentage of known nodes. The first two are identical to the parameters from the previous section, but the last one determines percentage of the known (labeled) nodes. Nine values of this parameter were used from 90% to 10% with a decrease of ten percent. The manufacturing company dataset required setting threshold on the contrary to the Enron dataset where it was not necessary. Additionally, to find the best utility score experiment was carried out for all calculated features, as well as the best Jaccard value and threshold were chosen from a range of different values. The hyperparameters search space is shown in Table A26 as well as the best ones for each model in Tables A27–A31.

Algorithm 1 Collective Classification Algorithm.

```
sort nodes descending by utility score
assign given percentage of top $v_i \in V$ to $V^L$ for each label
repeat
    //perform message passing
    for each edge $(v_i, v_j) \in E, v_i \in V^L, v_j \in V^{UK}$ do
        $c_{V_j}(l_{V_i}) \leftarrow c_{V_j}(l_{V_i}) + 1$
    end for
    //perform label update
    for each node $v_i \in V^{UK}$ do
        if $l_{v_i}$ is a majority class then
            $c_{V_i}(l_{V_i}) \leftarrow c_{V_i}(l_{V_i}) / threshold$
        end if
        if exists only one label with highest count for the node $v_i$ then
            $l_{v_i} \leftarrow l : \max_{l \in L} c_{v_i}(l)$
            assign $v_i$ to $V^L$
            $u_{v_i} \leftarrow 0$
        else
            $u_{v_i} \leftarrow u_{v_i} + 1$
        end if
        if the maximum value of $u_{v_i}$ has been reached then
            //get set of labels with the highest count
            $L_{max} \leftarrow l : \max_{l \in L} c_{v_i}(l)$
            //get label with the highest position in the hierarchy (smaller is higher)
            $l_{v_i} \leftarrow l : \min L_{max}$
            assign $v_i$ to $V^L$
            $u_{v_i} \leftarrow 0$
        end if
    end for
until stop condition
```

3. Results

The problem that is tackled can be considered as a binary classification for two groups of employees and multiclass classification for three groups. Therefore, f-score macro average measure was used to evaluate the solution in sight of the one metric which was needed to compare both results. This measure can handle the above cases, moreover, as was written in Reference [42] it copes well with the problem of unbalanced classes. The biggest advantage of this measure is the equal treatment of all classes which means that a result is not dominated by a majority class.

The results for the manufacturing company dataset are shown in Figures 3, 4 and 5. The f-score macro average for the randomly assigned labels is around 0.42 for two levels of the hierarchy and 0.24 for three levels of the hierarchy, in comparing the best result for the two levels was 0.7768 obtained by Random Forest and for the three levels 0.4737 achieved by Decision Tree. The much higher score obtained for two groups of classification can be explained by unbalanced classes. The classification of the three groups got worse results because of the small number of samples which was insufficient for the distinction between the two levels of management, even when oversampling was used. Furthermore, Random Forest got a slightly better results especially for two groups of employees. A strange phenomenon can be observed when a reduction of the most important features occasionally concludes with a better result; meaning that there could be some noise among the features which may affect the decision boundary. The potentially explanation of this phenomenon might be related to the problem described in Reference [2], so the observed alteration could be a result that the hierarchy, which arises from daily duties does not converge with company structure on paper. This inconsistency could be the source of some noise in the used features which has an influence on the obtained result; therefore, changing the network structure by eliminating some nodes, as well as removing the most important features, could result in moving a decision boundary. It is also noticeable that the parameter of a minimum employee activity also has impact on the classification but it is difficult to indicate the best value because no clear pattern is visible; however, most of the best results are obtained for a minimum activity greater than one month. The best results for two and three groups of employees was obtained by the collective classification algorithm which was able to classify nodes even if more than half of the labels were unknown.

Figures 6, 7, and 8 present the results for the Enron dataset which are similar to the results of the previous dataset. The result obtained by random labels was equal 0.49 for the two levels of the hierarchy and 0.33 for the three levels. The best f1-score for the supervised learning methods was achieved by Random Forest algorithm, and it was 0.8198 for the two hierarchy levels and 0.6423 for the three levels. The results of the collective classification algorithm where higher than the results of the standard classification if the knowledge of the node labels was over 70%. Below this value, the results were similar to standard classification, moreover, for three groups, if the knowledge of nodes fell below 40%, the results significantly deteriorated. It is visible that excessive reduction of features or known nodes leads to the results close to randomness. Furthermore, similarity of the results is important because shows that the presented solution works well for various organizational management models. In the manufacturing company the majority of the employees are regular clerical workers in contrast to the small management group. In the Enron dataset the situation is opposite, so the ratio between the first and second management level and regular employees is balanced.

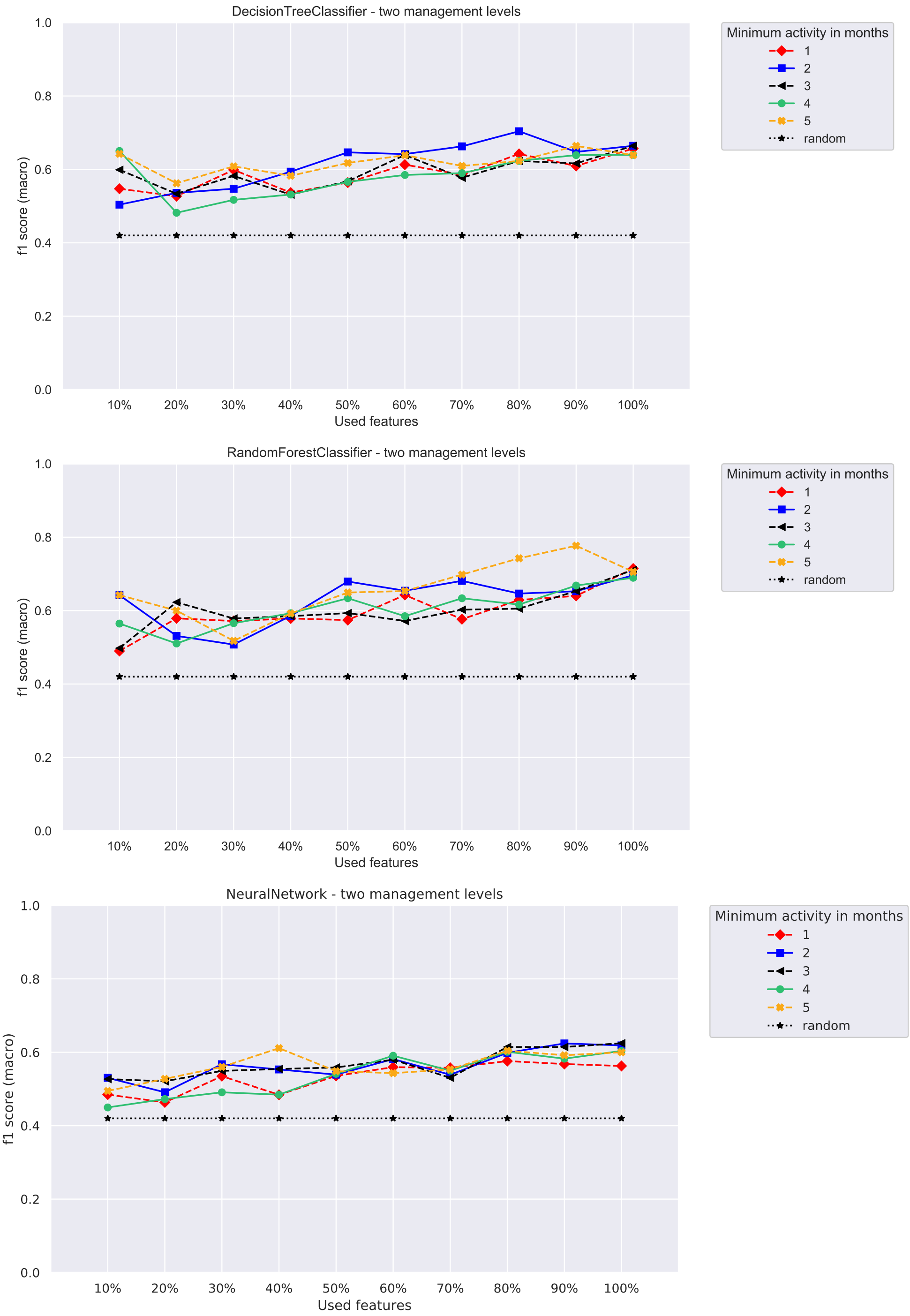
As a summary, the best results obtained by supervised learning algorithms are presented in Table 4. Moreover, all numerical results can be found in Appendix in Tables A1–A20.

Table 4. The best results obtained by supervised methods.

Dataset	Number of Levels	Algorithm	F1-Score	Min. Activity	% of Features
manufacturing company	2	Decision Tree	0.7039	2 months	80%
		Random Forest	0.7768	5 months	90%
		Neural Network	0.6247	3 months	100%
		SVM	0.6517	3 months	100%
	3	Decision Tree	0.4737	1 month	80%
		Random Forest	0.4575	2 months	60%
		Neural Network	0.4149	1 month	100%
		SVM	0.4622	4 months	50%
Enron	2	Decision Tree	0.7849	5 months	100%
		Random Forest	0.8198	5 months	100%
		Neural Network	0.7835	4 months	80%
		SVM	0.7855	5 months	50%
	3	Decision Tree	0.5827	5 months	50%
		Random Forest	0.6423	3 months	100%
		Neural Network	0.6107	4 months	70%
		SVM	0.6137	3 months	70%

Interesting conclusions about trained models can be drawn from the Figures A1–A12, which presents the importance of the features for models that used a set of all features. First of all, it should be noticed that the Decision Tree and Random Forest use many features; however, none of the features stand out significantly in terms of importance. Nevertheless, the clustering coefficient could be highlighted for the manufacturing company and indegree centrality for the Enron dataset because in many cases these features have slightly bigger importance than the others. For the

Neural Network and SVM algorithm the total number of cliques is visibly the best one for the both datasets. Moreover, sent and received neighborhood variability are also in some cases significant, therefore, it shows that not only network centrality measures but also features created from employees' behavior can be important in the classification task. A common element for all algorithms is the fact that for all parameters of the experiment, the worst feature is always the eigenvector centrality. Furthermore, Tables A27–A31 show the best utility score depending on different combinations of experiment parameters. Unlike supervised methods, it is difficult to identify the most discriminating feature for collective classification because a wide range of them is used as a utility score.


DecisionTreeClassifier - two management levels

RandomForestClassifier - two management levels

NeuralNetwork - two management levels

Figure 3. *Cont.*

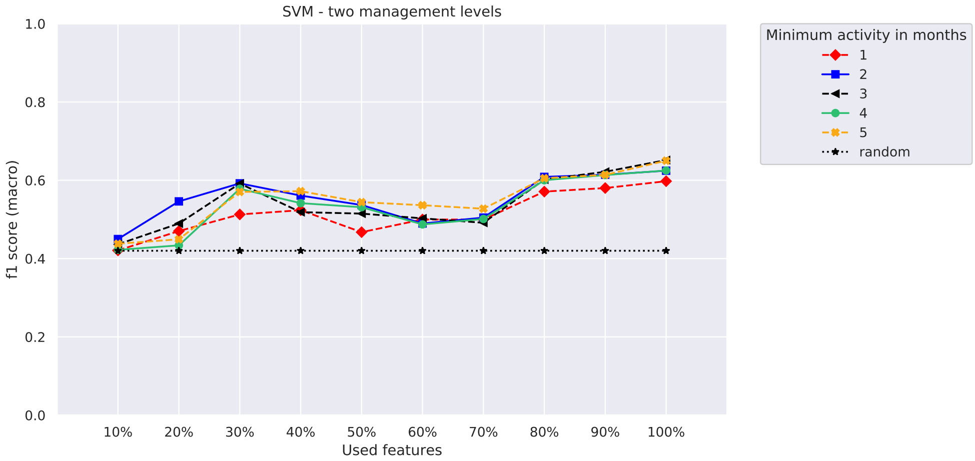


Figure 3. The result of the classification of two groups for the manufacturing company dataset.

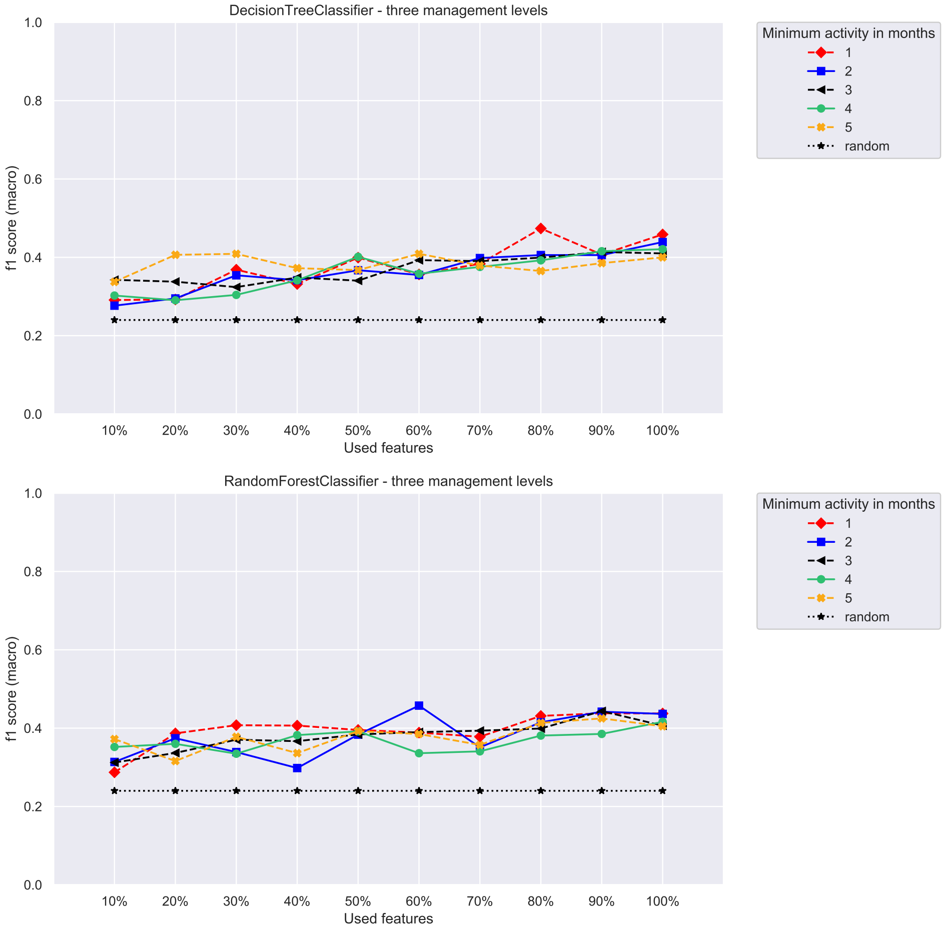


Figure 4. *Cont.*

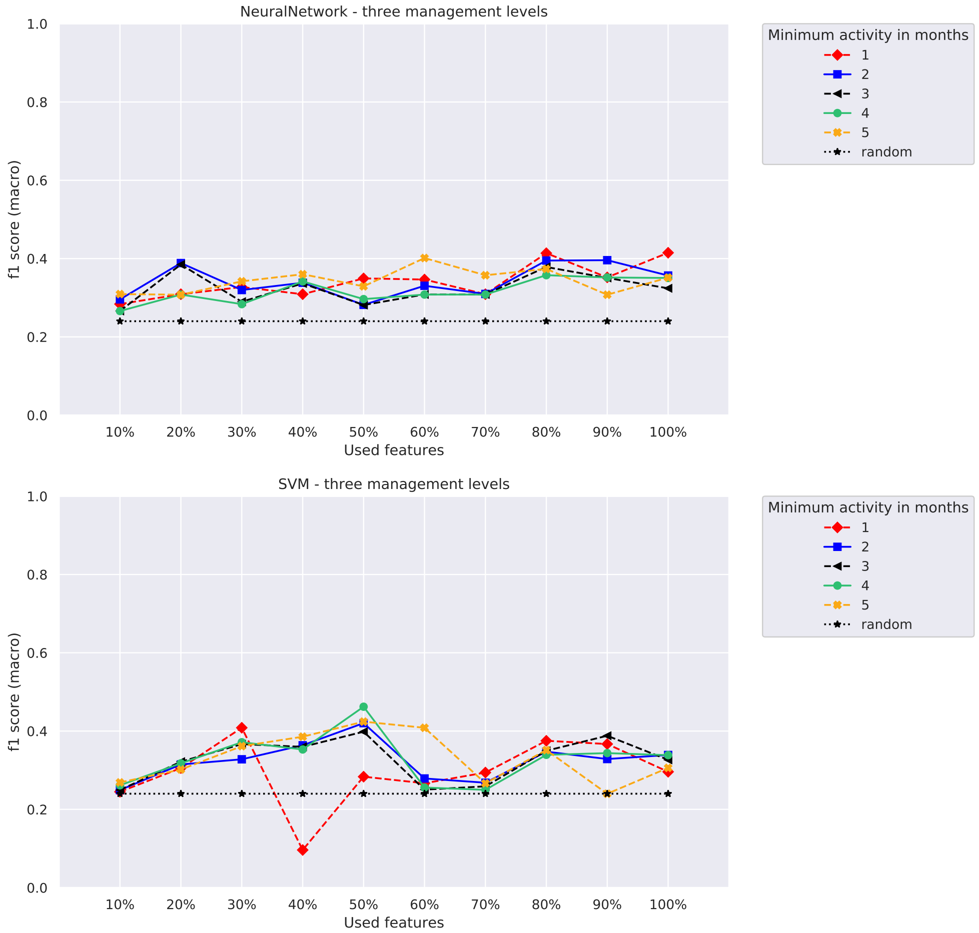


Figure 4. The result of the classification of three groups for the manufacturing company dataset.

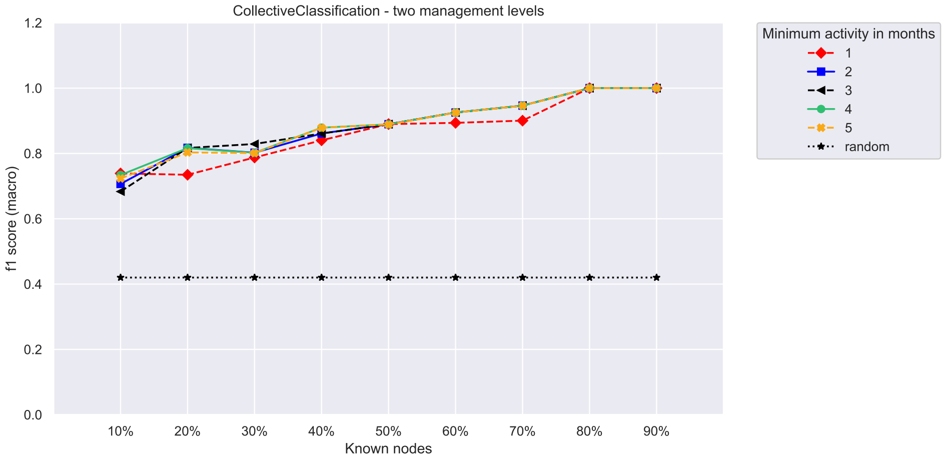


Figure 5. *Cont.*

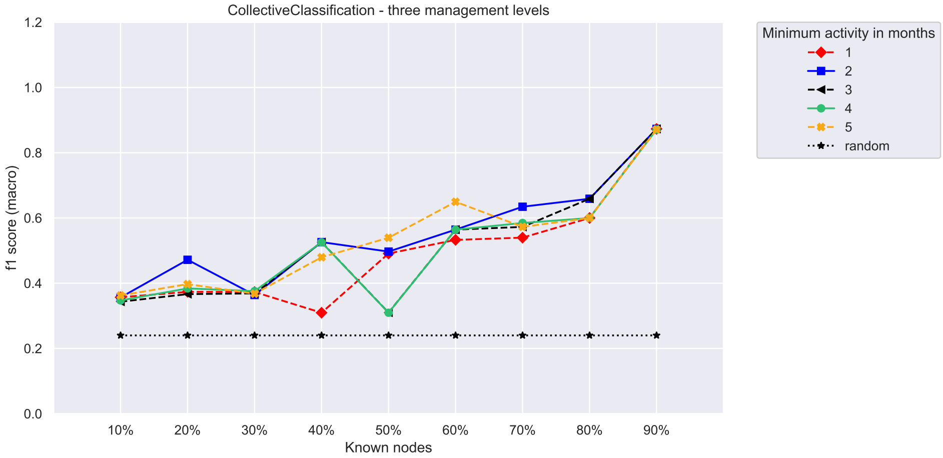


Figure 5. The result of the collective classification of three groups for the manufacturing company dataset.

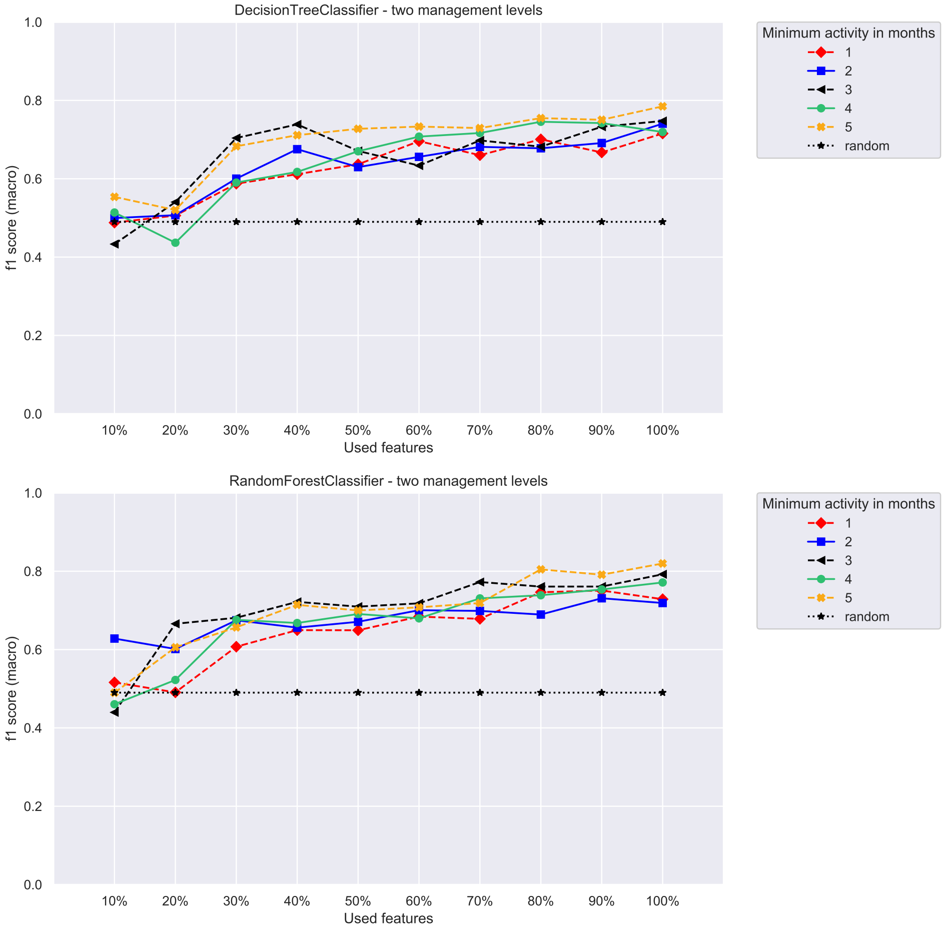


Figure 6. *Cont.*

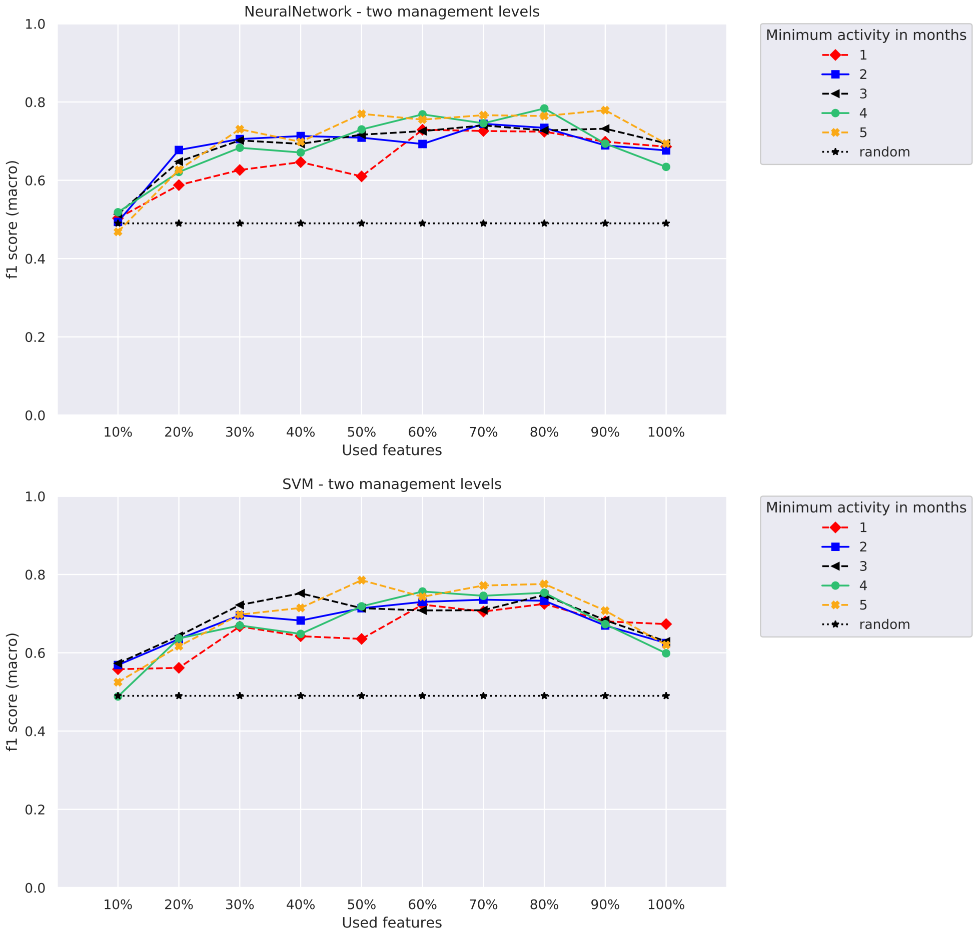


Figure 6. The result of the classification of two groups for the Enron dataset.

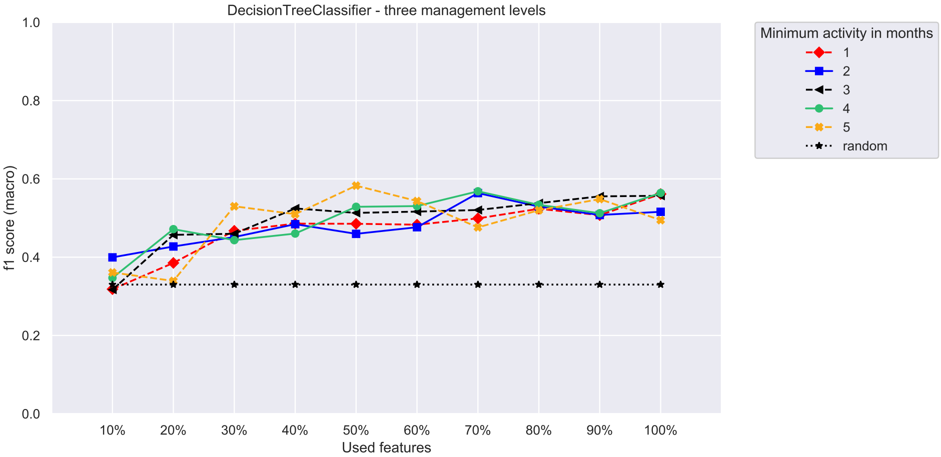


Figure 7. *Cont*.

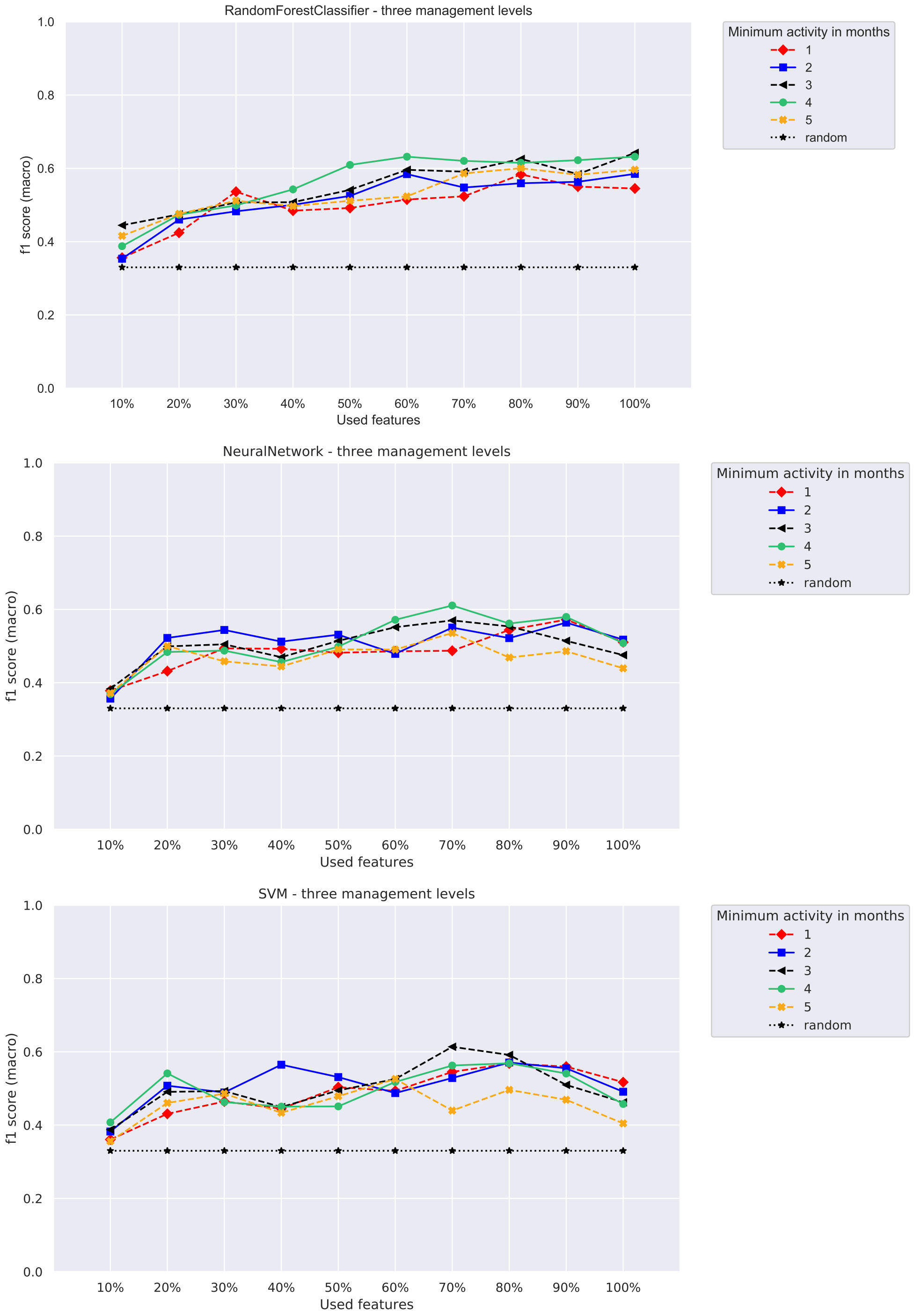


Figure 7. The result of the classification of three groups for the Enron company dataset.

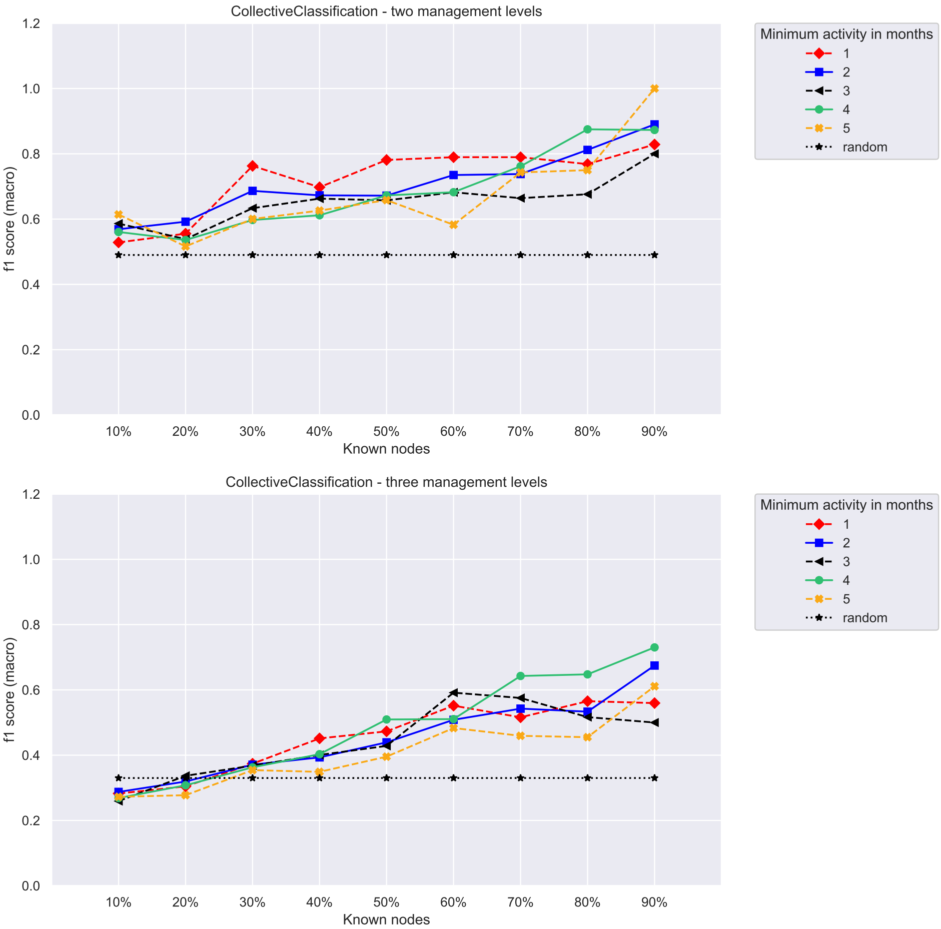


Figure 8. The result of the collective classification of three groups for the Enron company dataset.

4. Discussion

The comparison of the results for standard classification algorithms and collective classification show that the first of them cope better with balanced data such as the Enron datasets. The conducted experiments also present that Decision Tree and Random Forest, as well as Neural Network and SVM algorithm, have been able to obtain similar results. However, in an organizational environment, the possibility of the result interpretation could be highly appreciated, therefore the first two algorithms are a better solution if we would like to deeply understand the communication behavior on different organizational hierarchy levels. Furthermore, the presented collective classification algorithm obtained better results in the case of an unbalanced dataset. These results indicate that the graph algorithm is able to reduce impact of majority classes and predict well contrary to the standard classification algorithms. In addition, future research should examine if the above impact of unequal classes is also associated with some network hidden characteristics. Furthermore, it was presented that the minimum length of the period in which an employee was active influences the result, however the ideal value of the minimum activity depends on an analyzed dataset, as well as other parameters.

A network created from email communication may vary from organization to organization, especially regarding to the size of the company and different management models, for example: a big international company with many employees and a complicated hierarchy could create a social network with totally different properties than a small startup with a bunch of employees and a simple structure. Moreover, in some companies email communication could be one of many ways of passing messages, so a dataset of emails do not have to contain full information about the connection between employees in the company. These differences could cause some patterns of behavior assigned to a specific level of hierarchy, which does not have to appear in a constructed social network.

Future work should focus on the study of communication coming from different types of companies, moreover, further research should discover which organizational structures can provide the best results of classification task. Therefore, better results could be an implication of some hidden graph properties corresponding to the way which an organization is managed, so future studies also should focus on the examination of a network structure and revealing its characteristics. The biggest problem may be obtaining data for research due to the fact that internal communication of a company is confidential and has to be anonymized before being shared. Another interesting approach is the attempt to use graph embeddings instead of conventional features provided to supervised learning algorithms. This way the properties of nodes will be encoded in a form of vectors making them more suitable as direct input to algorithms. Regarding the collective classification, instead of analysing particular features as the input for utility score, latent Dirichlet allocation could be used to create a utility score combining features. Lastly, the reader would notice that the social network used in this study has been an aggregated one. This was mainly due to the fact that organizational structure of manufacturing company did not undergo any changes in the period covered by the dataset and in the case of Enron, the structure was inferred from e-mails and no other information was known. However, for applying proposed approach in organizations, it would be advised to verify the capabilities of temporal approach: both in the area of measures as well as networks.

Supplementary Materials: The manufacturing company email dataset alongside with corporate hierarchy has been published by the authors of this manuscript at Harvard Dataverse, see https://doi.org/10.7910/DVN/6Z3CGX. The Enron dataset is available from Reference [1].

Author Contributions: Conceptualization, R.M. and M.N.; methodology, R.M. and M.N.; software, M.N.; validation, R.M.; investigation, R.M. and M.N.; data curation, R.M. and M.N.; writing—original draft preparation, M.N. and R.M.; writing—review and editing, R.M.; visualization, M.N.; supervision, R.M.; project administration, R.M.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science Centre, Poland, project no. 2015/17/D/ST6/04046 as well as by the European Unions Horizon 2020 research and innovation programme under grant agreement no. 691152 (RENOIR) and the Polish Ministry of Science and Higher Education fund for supporting internationally co-financed projects in 2016–2019 (agreement no. 3628/H2020/2016/2).

Acknowledgments: Authors of this manuscript would like to thank to Piotr Bródka for his valuable remarks. Moreover, we would like to express our gratitude to the Reviewers of the manuscript for their precious feedback.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SNA	Social Network Analysis
IID	Independent and Identical Distribution
SVM	Support Vector Machine
L-BFGS	limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm

Appendix A

Table A1. F-score macro average for the Decision Tree classification of two levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.5472	0.5270	0.5987	0.5368	0.5641	0.6130	0.5850	0.6422	0.6092	**0.6568**
	2 months	0.5040	0.5360	0.5473	0.5938	0.6464	0.6417	0.6627	**0.7039**	0.6476	0.6642
	3 months	0.5989	0.5336	0.5822	0.5312	0.5678	0.6391	0.5772	0.6230	0.6162	**0.6642**
	4 months	**0.6502**	0.4818	0.5170	0.5315	0.5660	0.5848	0.5898	0.6234	0.6388	0.6398
	5 months	0.6423	0.5622	0.6083	0.5826	0.6174	0.6387	0.6094	0.6231	**0.6640**	0.6388

Table A2. F-score macro average for the Random Forest classification of two levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.4899	0.5789	0.5717	0.5786	0.5742	0.6426	0.5766	0.6290	0.6401	**0.7146**
	2 months	0.6421	0.5311	0.5075	0.5857	0.6793	0.6542	0.6811	0.6464	0.6528	**0.6963**
	3 months	0.4977	0.6228	0.5785	0.5849	0.5933	0.5718	0.6024	0.6054	0.6532	**0.7111**
	4 months	0.5647	0.5104	0.5657	0.5928	0.6337	0.5848	0.6336	0.6164	0.6683	**0.6895**
	5 months	0.6423	0.6000	0.5176	0.5923	0.6493	0.6533	0.6981	0.7425	**0.7768**	0.7057

Table A3. F-score macro average for the Neural Network classification of two levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.4848	0.4635	0.5349	0.4842	0.5356	0.5595	0.5577	**0.5760**	0.5680	0.5627
	2 months	0.5303	0.4910	0.5674	0.5532	0.5391	0.5814	0.5388	0.5986	**0.6241**	0.6187
	3 months	0.5270	0.5210	0.5493	0.5544	0.5586	0.5798	0.5304	0.6144	0.6144	**0.6247**
	4 months	0.4497	0.4725	0.4909	0.4845	0.5412	0.5906	0.5500	0.6013	0.5826	**0.6037**
	5 months	0.4946	0.5275	0.5601	**0.6112**	0.5486	0.5431	0.5534	0.6042	0.5919	0.6000

Table A4. F-score macro average for the SVM classification of two levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.4209	0.4703	0.5128	0.5236	0.4675	0.4998	0.4997	0.5711	0.5802	**0.5977**
	2 months	0.4499	0.5461	0.5922	0.5609	0.5368	0.4896	0.5047	0.6087	0.6142	**0.6245**
	3 months	0.4364	0.4896	0.5917	0.5186	0.5149	0.5027	0.4915	0.6018	0.6218	**0.6517**
	4 months	0.4223	0.4338	0.5783	0.5416	0.5312	0.4873	0.5012	0.6001	0.6134	**0.6250**
	5 months	0.4379	0.4493	0.5703	0.5722	0.5441	0.5363	0.5274	0.6053	0.6139	**0.6501**

Table A5. F-score macro average for the collective classification of two levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Known Nodes								
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Min. activity	1 month	0.7389	0.7347	0.7876	0.8402	0.8899	0.8937	0.9005	**1.0000**	**1.0000**
	2 months	0.7067	0.8172	0.8023	0.8613	0.8895	0.9253	0.9466	**1.0000**	**1.0000**
	3 months	0.6834	0.8168	0.8290	0.8610	0.8895	0.9253	0.9463	**1.0000**	**1.0000**
	4 months	0.7340	0.8163	0.8019	0.8790	0.8891	0.9253	0.9463	**1.0000**	**1.0000**
	5 months	0.7233	0.8031	0.8010	0.8786	0.8887	0.9250	0.9463	**1.0000**	**1.0000**

Table A6. F-score macro average for the Decision Tree classification of three levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.2910	0.2924	0.3685	0.3319	0.3992	0.3564	0.3835	**0.4737**	0.4071	0.4585
	2 months	0.2766	0.2950	0.3541	0.3416	0.3670	0.3550	0.3981	0.4058	0.4057	**0.4390**
	3 months	0.3425	0.3378	0.3239	0.3484	0.3404	0.3928	0.3903	0.3993	**0.4136**	0.4100
	4 months	0.3024	0.2904	0.3041	0.3413	0.4014	0.3582	0.3755	0.3923	0.4156	**0.4208**
	5 months	0.3373	0.4064	0.4089	0.3723	0.3674	**0.4091**	0.3790	0.3651	0.3852	0.3999

Table A7. F-score macro average for the Random Forest classification of three levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.2872	0.3867	0.4075	0.4065	0.3954	0.3886	0.3780	0.4312	**0.4380**	0.4374
	2 months	0.3137	0.3742	0.3389	0.2981	0.3834	**0.4575**	0.3510	0.4149	0.4420	0.4363
	3 months	0.3122	0.3367	0.3705	0.3667	0.3839	0.3899	0.3935	0.3990	**0.4435**	0.4064
	4 months	0.3520	0.3599	0.3347	0.3822	0.3917	0.3360	0.3407	0.3810	0.3852	**0.4163**
	5 months	0.3720	0.3160	0.3778	0.3362	0.3925	0.3850	0.3571	0.4133	**0.4249**	0.4048

Table A8. F-score macro average for the Neural Network classification of three levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.2838	0.3090	0.3273	0.3090	0.3494	0.3465	0.3090	0.4138	0.3520	**0.4149**
	2 months	0.2957	0.3886	0.3197	0.3383	0.2818	0.3306	0.3097	0.3950	**0.3959**	0.3567
	3 months	0.2662	**0.3848**	0.2898	0.3365	0.2814	0.3084	0.3084	0.3785	0.3506	0.3241
	4 months	0.2660	0.3081	0.2836	0.3417	0.2965	0.3081	0.3081	**0.3576**	0.3517	0.3505
	5 months	0.3093	0.3077	0.3419	0.3600	0.3291	**0.4014**	0.3574	0.3736	0.3077	0.3513

Table A9. F-score macro average for the SVM classification of three levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.2449	0.3055	**0.4085**	0.0962	0.2835	0.2667	0.2941	0.3752	0.3669	0.2960
	2 months	0.2493	0.3147	0.3279	0.3637	**0.4203**	0.2789	0.2681	0.3474	0.3286	0.3389
	3 months	0.2495	0.3218	0.3666	0.3597	**0.3984**	0.2504	0.2588	0.3497	0.3881	0.3265
	4 months	0.2615	0.3177	0.3715	0.3530	**0.4622**	0.2558	0.2499	0.3393	0.3437	0.3385
	5 months	0.2688	0.3018	0.3615	0.3856	**0.4241**	0.4084	0.2666	0.3514	0.2399	0.3060

Table A10. F-score macro average for the collective classification of three levels of the hierarchy in the manufacturing company dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Known Nodes								
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Min. activity	1 month	0.3570	0.3735	0.3731	0.3095	0.4910	0.5328	0.5397	0.6000	**0.8730**
	2 months	0.3564	0.4718	0.3640	0.5260	0.4969	0.5642	0.6344	0.6589	**0.8730**
	3 months	0.3433	0.3665	0.3689	0.5258	0.3099	0.5642	0.5726	0.6589	**0.8730**
	4 months	0.3472	0.3841	0.3757	0.5255	0.3095	0.5642	0.5846	0.6000	**0.8713**
	5 months	0.3626	0.3972	0.3678	0.4795	0.5394	0.6496	0.5726	0.6000	**0.8713**

Table A11. F-score macro average for the Decision Tree classification of two levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.4880	0.5061	0.5874	0.6116	0.6369	0.6959	0.6602	0.7007	0.6672	**0.7156**
	2 months	0.4997	0.5070	0.6003	0.6755	0.6297	0.6559	0.6814	0.6778	0.6912	**0.7406**
	3 months	0.4332	0.5405	0.7040	0.7388	0.6710	0.6336	0.6979	0.6818	0.7328	**0.7475**
	4 months	0.5138	0.4367	0.5903	0.6175	0.6702	0.7076	0.7168	**0.7455**	0.7421	0.7194
	5 months	0.5538	0.5201	0.6829	0.7113	0.7275	0.7332	0.7294	0.7548	0.7506	**0.7849**

Table A12. F-score macro average for the Random Forest classification of two levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.5162	0.4907	0.6073	0.6496	0.6493	0.6841	0.6784	0.7464	**0.7507**	0.7286
	2 months	0.6281	0.6015	0.6747	0.6560	0.6709	0.7005	0.6987	0.6894	**0.7310**	0.7189
	3 months	0.4397	0.6658	0.6812	0.7218	0.7096	0.7180	0.7724	0.7608	0.7609	**0.7922**
	4 months	0.4604	0.5224	0.6759	0.6678	0.6908	0.6797	0.7307	0.7387	0.7535	**0.7713**
	5 months	0.4897	0.6057	0.6567	0.7142	0.6998	0.7078	0.7188	0.8047	0.7911	**0.8198**

Table A13. F-score macro average for the Neural Network classification of two levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.5032	0.5877	0.6263	0.6464	0.6100	**0.7291**	0.7260	0.7241	0.6985	0.6858
	2 months	0.4938	0.6774	0.7053	0.7130	0.7089	0.6927	**0.7443**	0.7340	0.6892	0.6763
	3 months	0.5135	0.6478	0.7019	0.6930	0.7165	0.7256	**0.7402**	0.7273	0.7318	0.6944
	4 months	0.5184	0.6212	0.6831	0.6710	0.7299	0.7684	0.7456	**0.7835**	0.6937	0.6342
	5 months	0.4684	0.6268	0.7305	0.6982	0.7697	0.7550	0.7663	0.7642	**0.7790**	0.6936

Table A14. F-score macro average for the SVM classification of two levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.5580	0.5615	0.6673	0.6423	0.6354	0.7230	0.7054	**0.7249**	0.6802	0.6733
	2 months	0.5687	0.6350	0.6958	0.6824	0.7138	0.7300	**0.7356**	0.7329	0.6696	0.6249
	3 months	0.5731	0.6430	0.7221	**0.7519**	0.7139	0.7082	0.7087	0.7475	0.6833	0.6290
	4 months	0.4883	0.6367	0.6698	0.6483	0.7186	**0.7565**	0.7456	0.7534	0.6731	0.5987
	5 months	0.5248	0.6167	0.6975	0.7146	**0.7855**	0.7430	0.7718	0.7757	0.7077	0.6200

Table A15. F-score macro average for the collective classification of two levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Known Nodes								
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Min. activity	1 month	0.5285	0.5554	0.7629	0.6976	0.7812	0.7896	0.7896	0.7684	**0.8286**
	2 months	0.5689	0.5920	0.6865	0.6727	0.6718	0.7350	0.7380	0.8120	**0.8901**
	3 months	0.5863	0.5385	0.6334	0.6627	0.6573	0.6821	0.6641	0.6761	**0.8000**
	4 months	0.5605	0.5359	0.5972	0.6117	0.6725	0.6824	0.7618	**0.8750**	0.8730
	5 months	0.6141	0.5163	0.6003	0.6260	0.6581	0.5826	0.7429	0.7500	**1.0000**

Table A16. F-score macro average for the Decision Tree classification of three levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.3183	0.3853	0.4674	0.4857	0.4853	0.4830	0.4991	0.5236	0.5074	**0.5613**
	2 months	0.3995	0.4273	0.4514	0.4843	0.4596	0.4765	**0.5637**	0.5293	0.5078	0.5159
	3 months	0.3170	0.4570	0.4598	0.5245	0.5131	0.5164	0.5205	0.5380	0.5554	**0.5568**
	4 months	0.3471	0.4716	0.4435	0.4603	0.5287	0.5305	**0.5683**	0.5337	0.5123	0.5641
	5 months	0.3609	0.3393	0.5300	0.5099	**0.5827**	0.5434	0.4761	0.5199	0.5484	0.4946

Table A17. F-score macro average for the Random Forest classification of three levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.3562	0.4244	0.5364	0.4846	0.4919	0.5150	0.5237	**0.5835**	0.5502	0.5451
	2 months	0.3535	0.4604	0.4828	0.5005	0.5248	0.5843	0.5477	0.5594	0.5633	**0.5849**
	3 months	0.4447	0.4745	0.5070	0.5076	0.5413	0.5961	0.5910	0.6258	0.5842	**0.6423**
	4 months	0.3877	0.4735	0.4991	0.5427	0.6094	0.6318	0.6204	0.6151	0.6224	**0.6320**
	5 months	0.4158	0.4758	0.5117	0.4970	0.5116	0.5235	0.5862	**0.6004**	0.5826	0.5963

Table A18. F-score macro average for the Neural Network classification of three levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.3783	0.4316	0.4938	0.4925	0.4817	0.4857	0.4872	0.5439	**0.5721**	0.5104
	2 months	0.3566	0.5221	0.5439	0.5121	0.5310	0.4787	0.5504	0.5217	**0.5639**	0.5176
	3 months	0.3835	0.4990	0.5048	0.4696	0.5142	0.5514	**0.5700**	0.5535	0.5142	0.4754
	4 months	0.3688	0.4838	0.4875	0.4563	0.4983	0.5716	**0.6107**	0.5615	0.5795	0.5080
	5 months	0.3714	0.4998	0.4583	0.4443	0.4909	0.4904	**0.5356**	0.4687	0.4860	0.4393

Table A19. F-score macro average for the SVM classification of three levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Used Features									
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Min. activity	1 month	0.3607	0.4308	0.4644	0.4448	0.5035	0.4934	0.5452	**0.5682**	0.5596	0.5171
	2 months	0.3825	0.5075	0.4891	0.5648	0.5310	0.4874	0.5284	**0.5708**	0.5551	0.4910
	3 months	0.3874	0.4907	0.4929	0.4492	0.4946	0.5256	**0.6137**	0.5915	0.5097	0.4622
	4 months	0.4075	0.5411	0.4626	0.4505	0.4511	0.5177	0.5626	**0.5687**	0.5407	0.4576
	5 months	0.3559	0.4604	0.4859	0.4337	0.4787	**0.5259**	0.4397	0.4962	0.4691	0.4046

Table A20. F-score macro average for the collective classification of three levels of the hierarchy in the Enron dataset. The values which are bolded are the best results for a given minimum communication activity.

		Percentage of the Known Nodes								
		10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Min. activity	1 month	0.2826	0.3044	0.3739	0.4514	0.4730	0.5517	0.5160	**0.5654**	0.5596
	2 months	0.2874	0.3193	0.3708	0.3933	0.4393	0.5084	0.5426	0.5333	**0.6746**
	3 months	0.2593	0.3368	0.3681	0.4007	0.4284	**0.5915**	0.5753	0.5166	0.5000
	4 months	0.2687	0.3077	0.3628	0.4031	0.5094	0.5105	0.6427	0.6476	**0.7302**
	5 months	0.2725	0.2774	0.3542	0.3489	0.3954	0.4832	0.4593	0.4551	**0.6111**

Table A21. Hyperparameter search space for supervised learning algorithms.

		Values
Decision Tree	max_depth	1, 2, 3, ..., 20
	max_features	1, 2, 3, ..., 16 (manufacturing company); 1, 2, 3, ..., 15 (Enron)
Random Forest	max_depth	1, 2, 3, ..., 20
	max_features	1, 2, 3, ..., 16 (manufacturing company); 1, 2, 3, ..., 15 (Enron)
	n_estimators	1, 2, 4, 8, 16, 32, 64, 100, 200
Neural Network	alpha	0.0001, 0.001, 0.01, 1
	hidden_layer_sizes	(13), (9), (4), (13, 9), (13, 4), (9, 4), (13, 9, 4), (4, 9, 4), (9, 13, 9), (9, 13, 4), (4, 9, 13, 9, 4)
SVM	degree	3, 4, 5, 6, 7, 8, 9, 10, 15, 20
	C	1, 2, 3, 4, 5, 10, 15, 20, 25, 50, 100, 150, 200

Table A22. The best values of the hyperparameters for the models which uses the full set of features for the manufacturing company dataset with two hierarchy levels.

		Min. Activity				
		1 month	**2 months**	**3 months**	**4 months**	**5 months**
Decision Tree	max_depth	4	8	8	10	5
	max_features	3	2	1	9	1
Random Forest	max_depth	8	6	6	10	6
	max_features	2	2	12	3	2
	n_estimators	32	16	16	16	8
Neural Network	alpha	0.0001	0.01	0.0001	0.001	0.01
	hidden_layer_sizes	(13, 4)	(13)	(13, 9, 4)	(13, 9)	(13, 9)
SVM	degree	20	20	20	15	20
	C	150	200	200	200	200

Table A23. The best values of the hyperparameters for the models which uses the full set of features for the manufacturing company dataset with three hierarchy levels.

		Min. Activity				
		1 Month	**2 Months**	**3 Months**	**4 Months**	**5 Months**
Decision Tree	max_depth	10	7	12	13	8
	max_features	1	8	2	1	8
Random Forest	max_depth	7	5	7	10	10
	max_features	3	11	14	6	7
	n_estimators	2	4	16	32	8
Neural Network	alpha	0.001	0.0001	0.01	0.01	0.001
	hidden_layer_sizes	(9, 13, 4)	(4, 9, 13, 9, 4)	(4, 9, 13, 9, 4)	(13, 9, 4)	(13, 9, 4)
SVM	degree	15	20	20	20	20
	C	200	200	200	200	150

Table A24. The best values of the hyperparameters for the models which uses the full set of features for the Enron dataset with two hierarchy levels.

		Min. Activity				
		1 Month	**2 Months**	**3 Months**	**4 Months**	**5 Months**
Decision Tree	max_depth	2	8	7	3	2
	max_features	1	9	3	4	12
Random Forest	max_depth	2	5	3	9	4
	max_features	3	7	5	2	2
	n_estimators	4	2	2	16	2
Neural Network	alpha	0.01	0.001	0.0001	0.001	0.0001
	hidden_layer_sizes	(13, 9)	(4, 9, 4)	(4, 9, 4)	(4, 9, 13, 9, 4)	(9)
SVM	degree	6	3	7	3	6
	C	20	10	20	100	50

Table A25. The best values of the hyperparameters for the models which uses the full set of features for the Enron dataset with three hierarchy levels.

		Min. Activity				
		1 Month	**2 Months**	**3 Months**	**4 Months**	**5 Months**
Decision Tree	max_depth	5	4	5	8	4
	max_features	7	8	11	9	13
Random Forest	max_depth	2	4	7	2	3
	max_features	2	12	2	14	14
	n_estimators	2	4	8	2	16
Neural Network	alpha	0.001	0.01	0.01	0.0001	0.001
	hidden_layer_sizes	(4, 3)	(13)	(13)	(9)	(13, 9, 4)
SVM	degree	9	3	3	6	3
	C	5	20	200	150	4

Table A26. Hyperparameter search space for collective classification.

Parameter	Values
utility score	all features from Table 3
threshold	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Jaccard value	0.7, 0.8, 0.9, 0.99

Table A27. The best values of the hyperparameters for the collective classification algorithm for the manufacturing company dataset with two hierarchy levels (1 to 3 months of minimum activity).

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
1 month	90%	hubs centrality clustering coefficient overtime	4 5 4	0.7 0.7 0.7
	80%	hubs centrality	4	0.9
	70%	indegree centrality the total numbers of cliques the biggest clique	4 4 4	0.7 0.7 0.7
	60%	indegree centrality outdegree centrality the total numbers of cliques the biggest clique	4 4 4 4	0.7 0.7 0.7 0.7
	50%	indegree centrality outdegree centrality	4 4	0.7 0.7
	40%	the total numbers of cliques	4	0.7
	30%	page rank centrality	4	0.7
	20%	betweenness centrality	5	0.7
	10%	the number of weekends worked	4	0.7
2 months	90%	closeness centrality hubs centrality clustering coefficient	4 4 5	0.7 0.7 0.7
	80%	hubs centrality the biggest clique	4 4	0.9 0.7
	70%	indegree centrality the total numbers of cliques the biggest clique	4 4 4	0.7 0.7 0.7
	60%	indegree centrality outdegree centrality the total numbers of cliques	4 4 4	0.7 0.7 0.7
	50%	indegree centrality	4	0.9
	40%	the total numbers of cliques	4	0.7
	30%	authorities centrality	4	0.9
	20%	betweenness centrality	5	0.7
	10%	the number of weekends worked	4	0.7

Table A27. *Cont.*

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
3 months	90%	closeness centrality hubs centrality clustering coefficient	4 4 5	0.7 0.7 0.7
	80%	hubs centrality the biggest clique	4 4	0.9 0.7
	70%	indegree centrality the total numbers of cliques the biggest clique	4 4 4	0.7 0.7 0.7
	60%	indegree centrality outdegree centrality the total numbers of cliques	4 4 4	0.7 0.7 0.7
	50%	received neighborhood variability	4	0.7
	40%	the total numbers of cliques	4	0.7
	30%	eigenvector centrality	4	0.7
	20%	betweenness centrality	5	0.7
	10%	eigenvector centrality	4	0.7

Table A28. The best values of the hyperparameters for the collective classification algorithm for the manufacturing company dataset with two hierarchy levels (4 to 5 months of minimum activity).

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
4 month	90%	outdegree centrality eigenvector centrality hubs centrality clustering coefficient	3 3 4 5	0.7 0.7 0.7 0.7
	80%	hubs centrality the biggest clique	4 4	0.9 0.7
	70%	indegree centrality the total numbers of cliques the biggest clique	4 4 4	0.7 0.7 0.7
	60%	outdegree centrality the total numbers of cliques	4 4	0.7 0.7
	50%	the biggest clique received neighborhood variability	4 4	0.99 0.7
	40%	the total numbers of cliques	4	0.7
	30%	authorities centrality	4	0.9
	20%	betweenness centrality	5	0.7
	10%	authorities centrality	4	0.7

Table A28. *Cont.*

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
5 months	90%	outdegree centrality betweenness centrality closeness centrality hubs centrality clustering coefficient	3 3 4 4 5	0.7 0.7 0.7 0.7 0.7
	80%	hubs centrality the biggest clique	4 4	0.9 0.7
	70%	indegree centrality the total numbers of cliques the biggest clique	4 4 4	0.7 0.7 0.7
	60%	the total numbers of cliques	4	0.7
	50%	the biggest clique received neighborhood variability	4 4	0.99 0.7
	40%	the total numbers of cliques	4	0.7
	30%	authorities centrality	4	0.7
	20%	betweenness centrality	5	0.7
	10%	the number of weekends worked	4	0.7

Table A29. The best values of the hyperparameters for the collective classification algorithm for the manufacturing company dataset with three hierarchy levels.

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
1 month	90%	the number of weekends worked	7	0.7
	80%	overtime	4	0.7
	70%	overtime	5	0.7
	60%	hubs centrality	5	0.7
	50%	the number of weekends worked	5	0.7
	40%	closeness centrality clustering coefficient the number of weekends worked	5 2 4	0.7 0.7 0.7
	30%	closeness centrality	2	0.7
	20%	received neighborhood variability	3	0.7
	10%	the biggest clique	2	0.7

Table A29. *Cont.*

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
2 months	90%	the number of weekends worked	7	0.7
	80%	overtime	5	0.7
	70%	betweenness centrality	5	0.7
	60%	hubs centrality	5	0.7
	50%	the number of weekends worked	5	0.7
	40%	general neighborhood variability	5	0.7
	30%	closeness centrality	2	0.7
	20%	clustering coefficient	3	0.7
	10%	the biggest clique	2	0.7
3 months	90%	the number of weekends worked	7	0.7
	80%	overtime	5	0.7
	70%	hubs centrality	5	0.7
	60%	hubs centrality	5	0.7
	50%	hubs centrality	4	0.7
		clustering coefficient	2	0.7
		the number of weekends worked	3	0.7
	40%	general neighborhood variability	5	0.7
	30%	hubs centrality	2	0.7
	20%	received neighborhood variability	3	0.7
	10%	page rank centrality	4	0.7
		the number of weekends worked	5	0.7
4 months	90%	the number of weekends worked	7	0.7
	80%	betweenness centrality	4	0.7
		overtime	4	0.7
	70%	betweenness centrality	4	0.7
	60%	hubs centrality	5	0.7
	50%	closeness centrality	5	0.7
		eigenvector centrality	4	0.7
		hubs centrality	4	0.7
		clustering coefficient	2	0.7
		the number of weekends worked	2	0.7
	40%	general neighborhood variability	5	0.7
	30%	hubs centrality	2	0.7
	20%	received neighborhood variability	3	0.7
	10%	page rank centrality	4	0.7
		the number of weekends worked	5	0.7
5 months	90%	the number of weekends worked	7	0.7
	80%	betweenness centrality	4	0.7
	70%	eigenvector centrality	5	0.7
	60%	betweenness centrality	6	0.7
	50%	sent neighborhood variability	5	0.7
	40%	betweenness centrality	6	0.7
	30%	hubs centrality	2	0.7
	20%	received neighborhood variability	3	0.7
	10%	the number of weekends worked	5	0.7

Table A30. The best values of the hyperparameters for the collective classification algorithm for the Enron dataset with two hierarchy levels.

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
1 month	90%	page rank centrality	1	0.7
	80%	authorities centrality	1	0.7
	70%	authorities centrality	1	0.7
	60%	indegree centrality	1	0.8
	50%	the biggest clique	1	0.7
	40%	indegree centrality	1	0.7
	30%	indegree centrality	1	0.7
	20%	general neighborhood variability	1	0.7
	10%	closeness centrality	1	0.7
2 months	90%	received neighborhood variability	1	0.7
	80%	closeness centrality hubs centrality	1 1	0.7 0.7
	70%	closeness centrality	1	0.7
	60%	the biggest clique	1	0.7
	50%	the biggest clique	1	0.7
	40%	indegree centrality	1	0.7
	30%	indegree centrality	1	0.7
	20%	general neighborhood variability	1	0.99
	10%	general neighborhood variability	1	0.7
3 months	90%	hubs centrality	1	0.7
	80%	authorities centrality	1	0.7
	70%	authorities centrality	1	0.7
	60%	the biggest clique	1	0.99
	50%	the biggest clique	1	0.7
	40%	indegree centrality	1	0.7
	30%	indegree centrality	1	0.7
	20%	received neighborhood variability	1	0.7
	10%	general neighborhood variability	1	0.7
4 months	90%	closeness centrality hubs centrality	1 1	0.7 0.7
	80%	indegree centrality	1	0.7
	70%	indegree centrality	1	0.99
	60%	the biggest clique	1	0.8
	50%	betweenness centrality	1	0.7
	40%	betweenness centrality	1	0.7
	30%	page rank centrality	1	0.7
	20%	general neighborhood variability	1	0.99
	10%	eigenvector centrality	1	0.7

Table A30. *Cont.*

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
5 months	90%	authorities centrality hubs centrality	1 1	0.7 0.7
	80%	indegree centrality	1	0.7
	70%	outdegree centrality	1	0.7
	60%	outdegree centrality	1	0.7
	50%	betweenness centrality	1	0.7
	40%	betweenness centrality	1	0.7
	30%	betweenness centrality	1	0.7
	20%	betweenness centrality	1	0.7
	10%	the biggest clique	1	0.7

Table A31. The best values of the hyperparameters for the collective classification algorithm for the Enron dataset with three hierarchy levels.

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
1 month	90%	the biggest clique	1	0.7
	80%	hubs centrality	1	0.7
	70%	page rank centrality	1	0.7
	60%	outdegree centrality	1	0.7
	50%	eigenvector centrality	1	0.7
	40%	the total numbers of cliques	1	0.7
	30%	indegree centrality	1	0.7
	20%	eigenvector centrality	1	0.7
	10%	general neighborhood variability	1	0.7
2 months	90%	received neighborhood variability	1	0.7
	80%	hubs centrality	1	0.7
	70%	eigenvector centrality	1	0.7
	60%	outdegree centrality	1	0.7
	50%	outdegree centrality	1	0.7
	40%	hubs centrality	1	0.7
	30%	indegree centrality	1	0.7
	20%	authorities centrality	1	0.7
	10%	the total numbers of cliques	1	0.99

Table A31. *Cont.*

		Hyperparameters		
Min. Activity	**% of the Known Nodes**	**Utility Score**	**Threshold**	**Jaccard Value**
3 months	90%	the biggest clique	1	0.7
	80%	hubs centrality	1	0.7
	70%	closeness centrality	1	0.7
	60%	outdegree centrality	1	0.7
	50%	the biggest clique	1	0.7
	40%	authorities centrality	1	0.7
	30%	outdegree centrality	1	0.7
	20%	sent neighborhood variability	1	0.7
	10%	the total numbers of cliques	1	0.9
4 months	90%	closeness centrality	1	0.7
	80%	authorities centrality	1	0.7
	70%	outdegree centrality	1	0.7
	60%	betweenness centrality	1	0.7
	50%	outdegree centrality	1	0.7
	40%	general neighborhood variability	1	0.7
	30%	betweenness centrality	1	0.7
	20%	general neighborhood variability	1	0.7
	10%	received neighborhood variability	1	0.7
5 months	90%	the biggest clique	1	0.7
	80%	the biggest clique	1	0.8
	70%	outdegree centrality	1	0.7
	60%	closeness centrality	1	0.7
	50%	the biggest clique	1	0.7
	40%	the number of weekends worked	1	0.7
	30%	the number of weekends worked	1	0.8
	20%	outdegree centrality	1	0.7
	10%	clustering coefficient	1	0.7

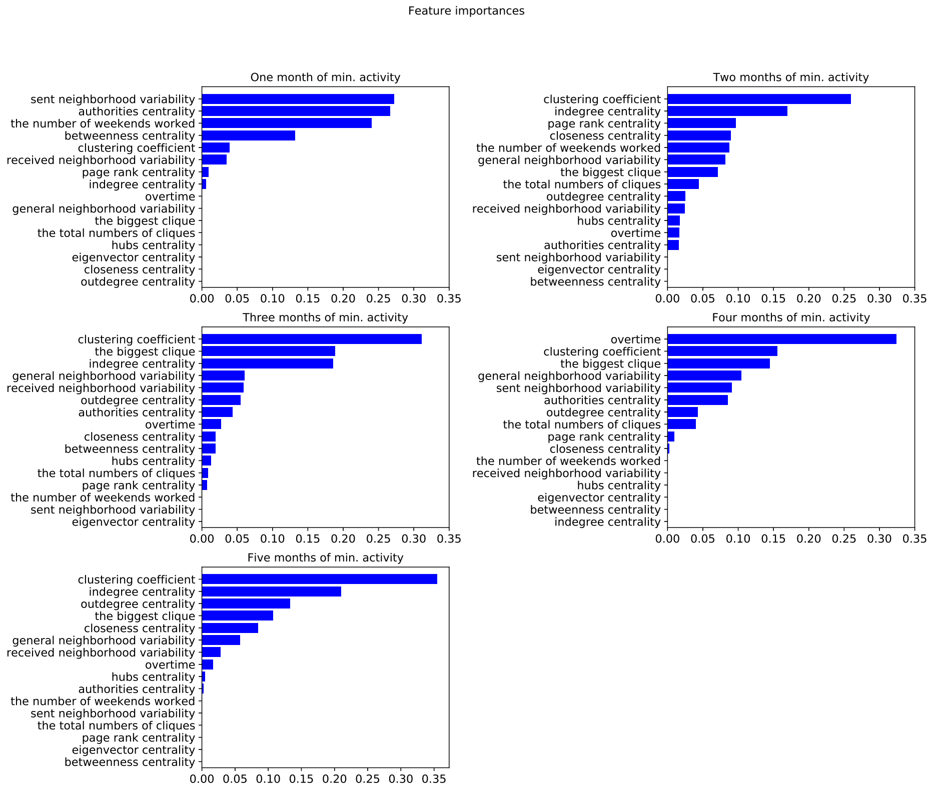


Figure A1. Features importance (Gini importance) for the Decision Tree which uses the full set of features for the manufacturing company dataset with two levels of the hierarchy.

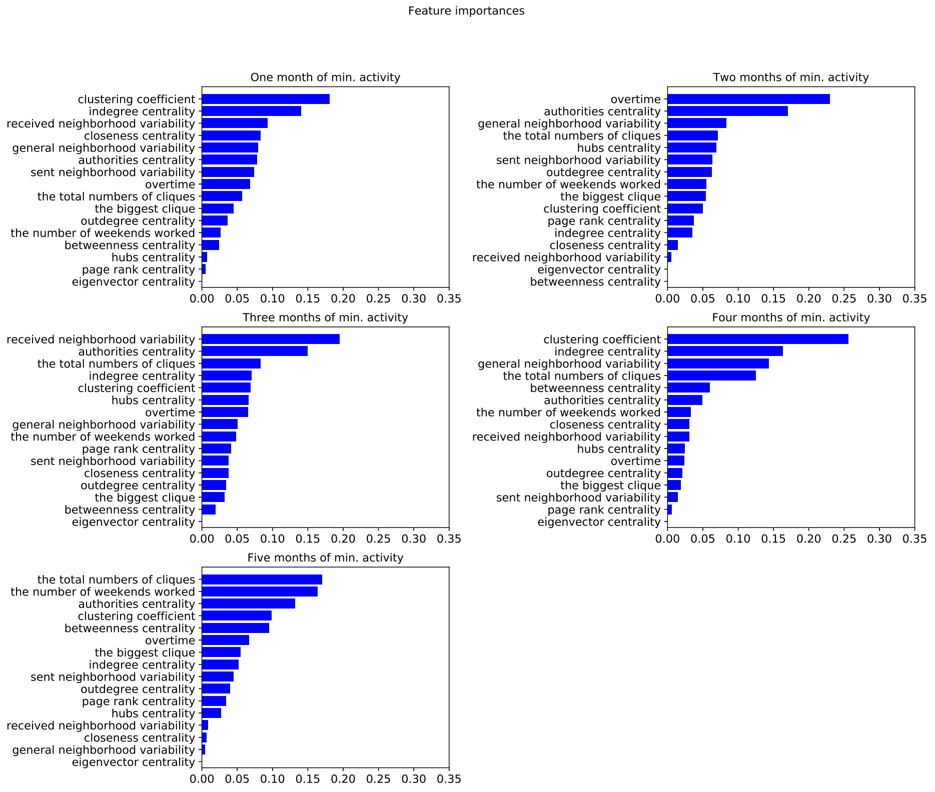


Figure A2. Features importance (Gini importance) for the Decision Tree which uses the full set of features for the manufacturing company dataset with three levels of the hierarchy.

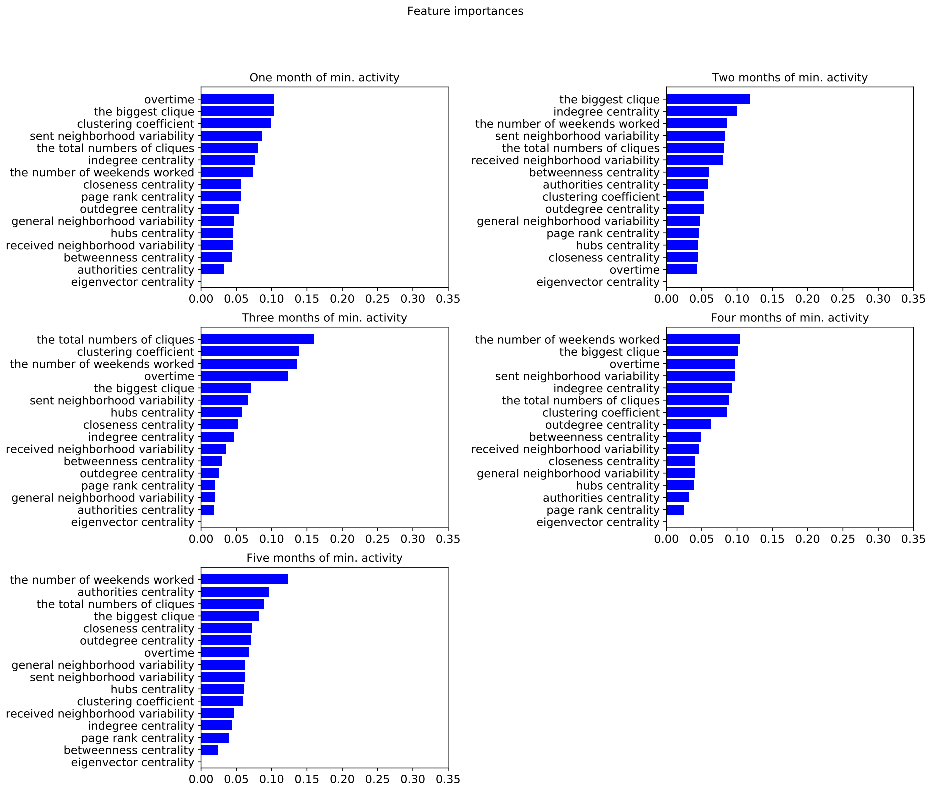


Figure A3. Features importance (Gini importance) for the Random Forest which uses the full set of features for the manufacturing company dataset with two levels of the hierarchy.

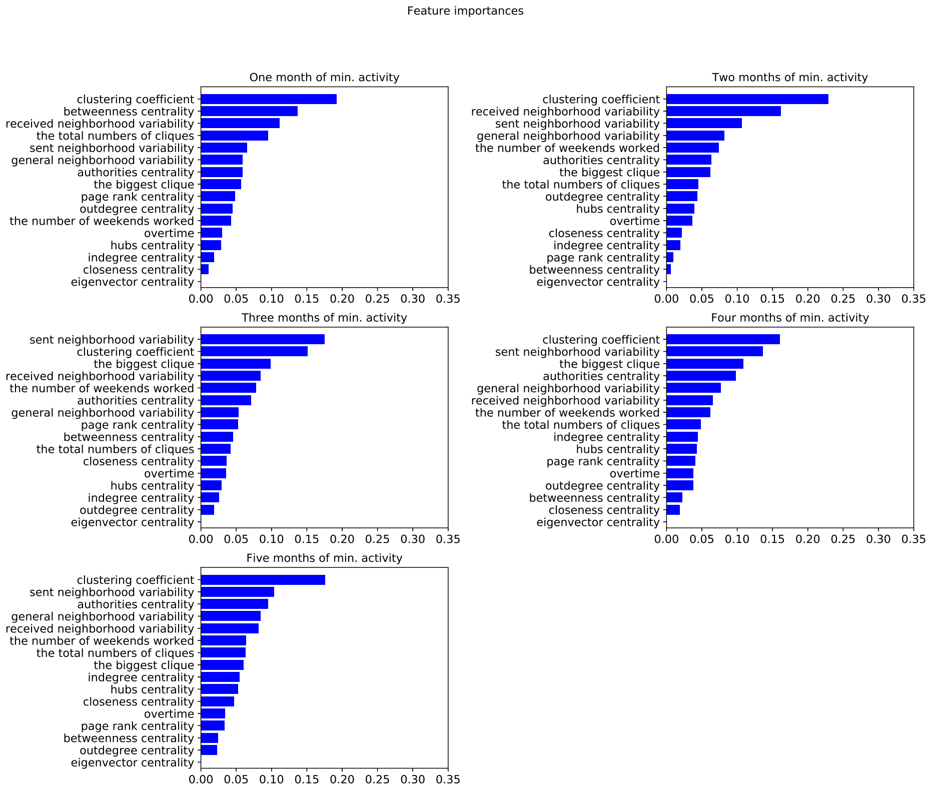


Figure A4. Features importance (Gini importance) for the Random Forest which uses the full set of features for the manufacturing company dataset with three levels of the hierarchy.

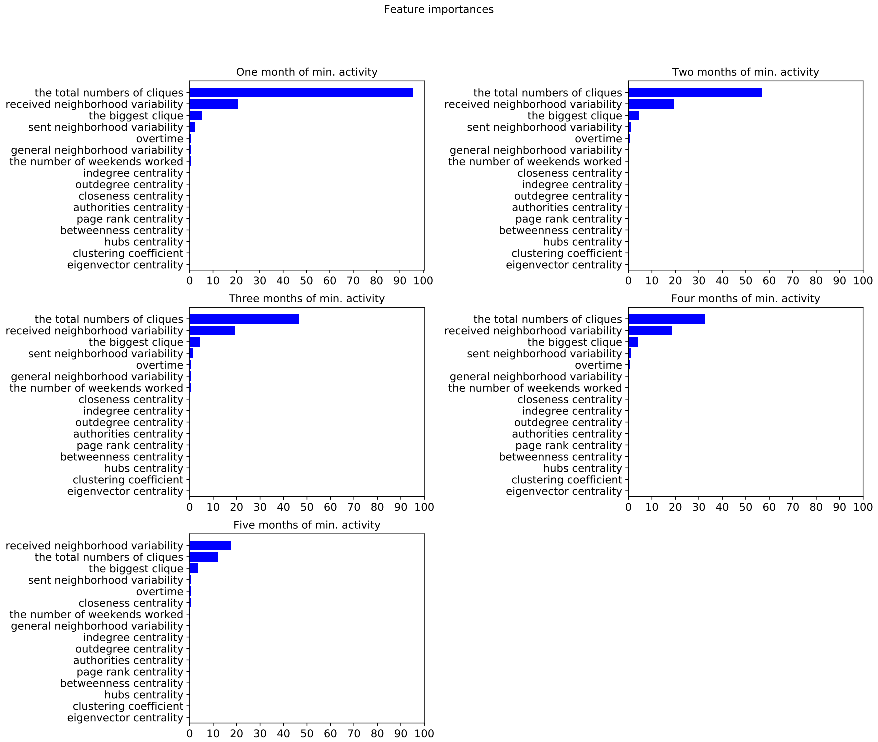


Figure A5. Features importance (based on univariate feature selection method which uses chi-squared test) for the Neural Network and SVM which use the full set of features for the manufacturing company dataset with two levels of the hierarchy.

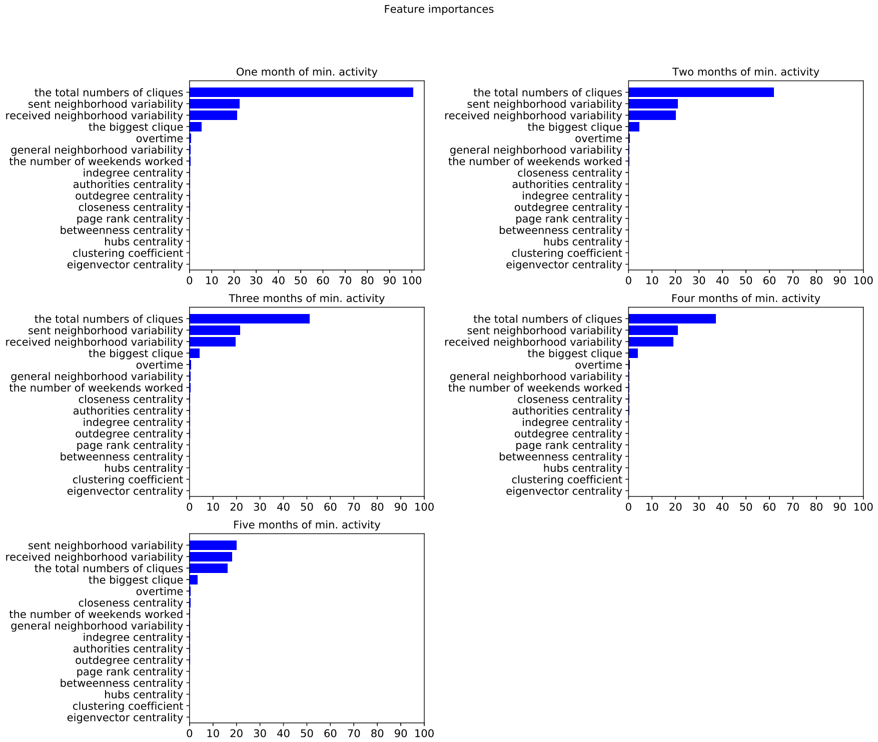


Figure A6. Features importance (based on univariate feature selection method which uses chi-squared test) for the Neural Network and SVM which use the full set of features for the manufacturing company dataset with three levels of the hierarchy.

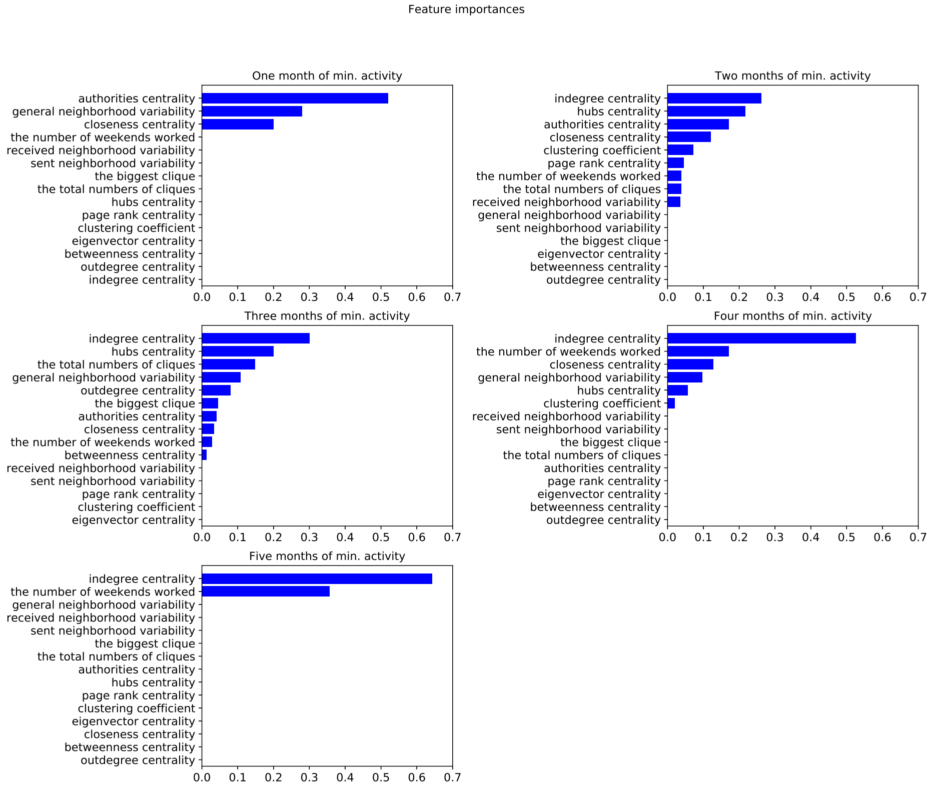


Figure A7. Features importance (Gini importance) for the Decision Tree which uses the full set of features for the Enron dataset with two levels of the hierarchy.

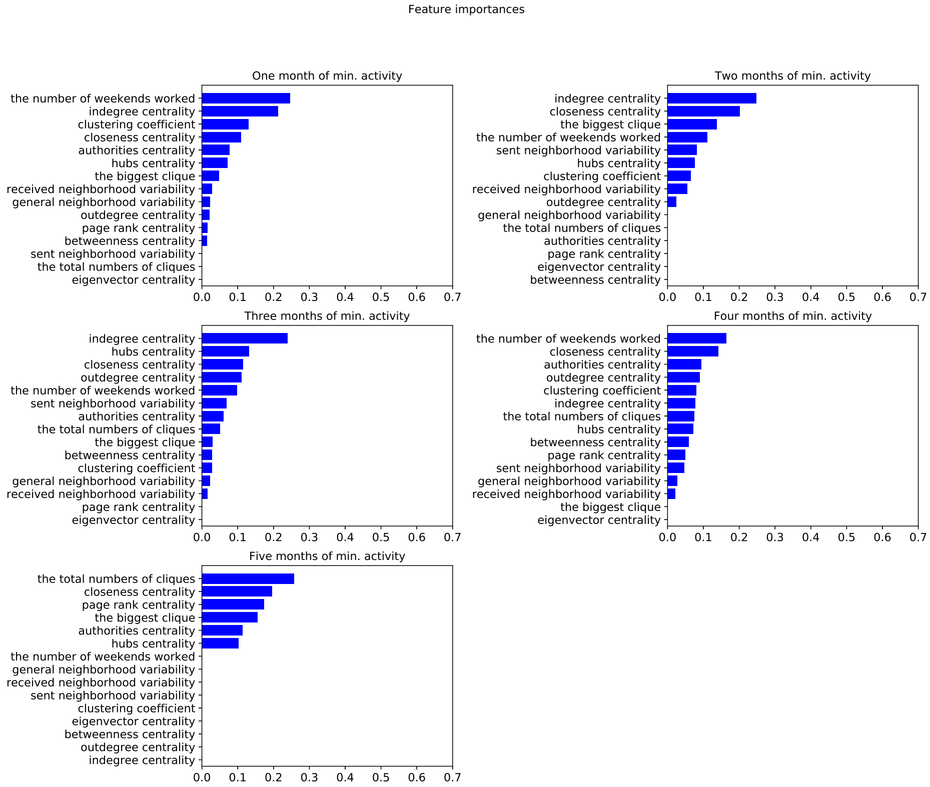


Figure A8. Features importance (Gini importance) for the Decision Tree which uses the full set of features for the Enron dataset with three levels of the hierarchy.

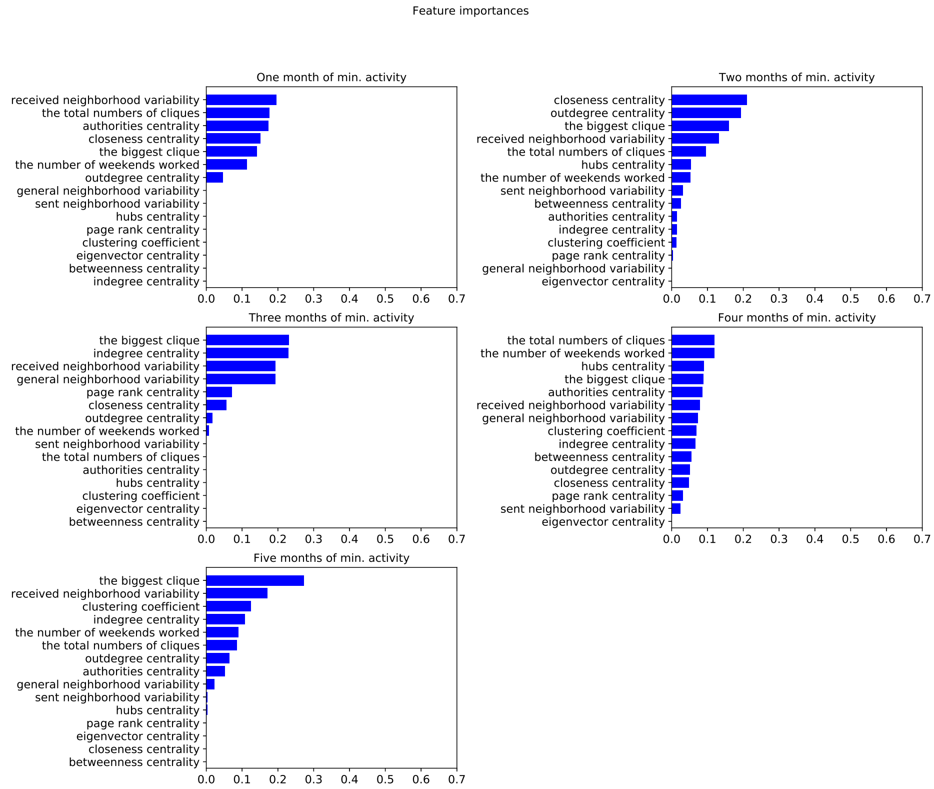


Figure A9. Features importance (Gini importance) for the Random Forest which uses the full set of features for the Enron dataset with two levels of the hierarchy.

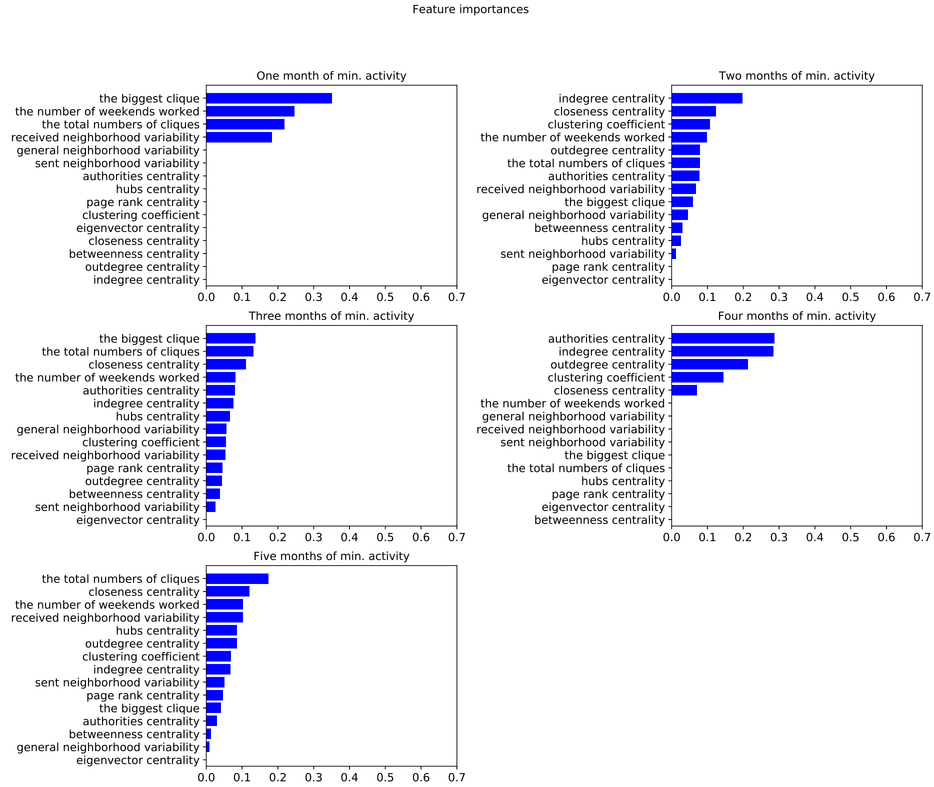


Figure A10. Features importance (Gini importance) for the Random Forest which uses the full set of features for the Enron dataset with three levels of the hierarchy.

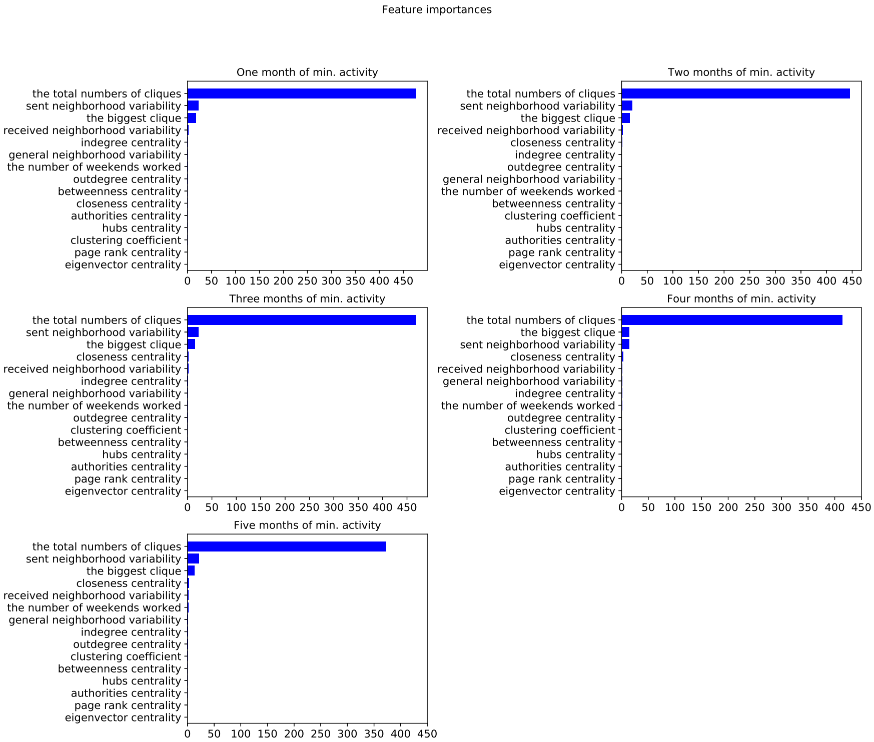


Figure A11. Features importance (based on univariate feature selection method which uses chi-squared test) for the Neural Network and SVM which use the full set of features for the Enron dataset with two levels of the hierarchy.

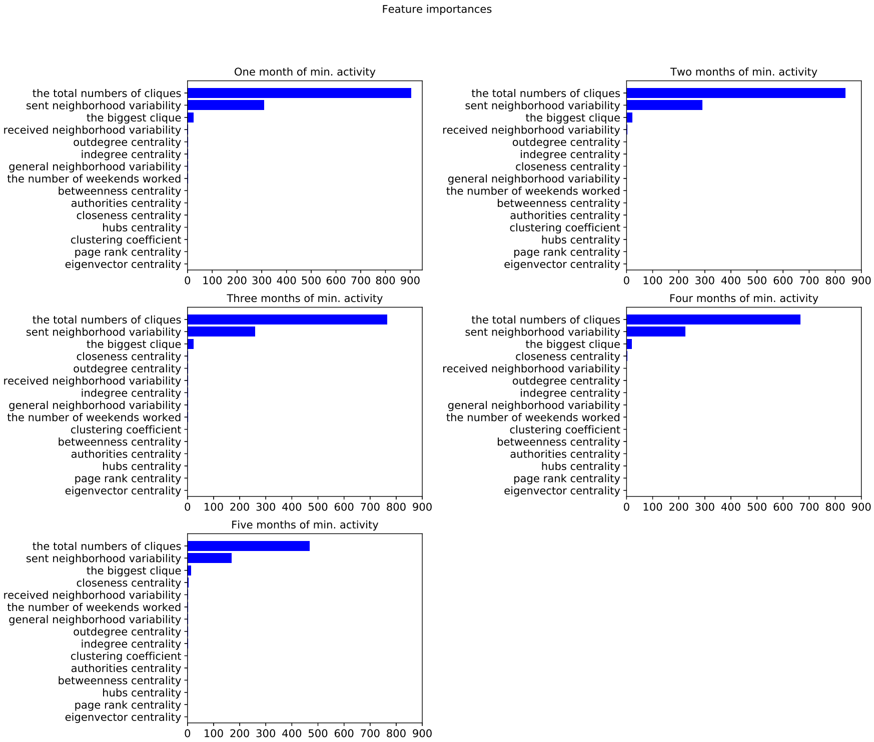


Figure A12. Features importance (based on univariate feature selection method which uses chi-squared test) for the Neural Network and SVM which use the full set of features for the Enron dataset with three levels of the hierarchy.

References

1. McCallum, A.; Wang, X.; Corrada-Emmanuel, A. Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email. *J. Artif. Intell. Res.* **2007**, *30*, 249–272. [CrossRef]
2. Palus, S.; Kazienko, P.; Michalski, R. Evaluation of Corporate Structure Based on Social Network Analysis. In *Social Development and High Technology Industries*; IGI Global: Hershey, PA, USA, 2012; pp. 58–69. [CrossRef]
3. Michalski, R.; Palus, S.; Kazienko, P. Matching Organizational Structure and Social Network Extracted from Email Communication. In *Business Information Systems*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 197–206. [CrossRef]
4. Creamer, G.; Rowe, R.; Hershkop, S.; Stolfo, S.J. Segmentation and Automated Social Hierarchy Detection through Email Network Analysis. In *Advances in Web Mining and Web Usage Analysis*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 40–58. [CrossRef]
5. Fire, M.; Puzis, R. Organization Mining Using Online Social Networks. *Networks Spat. Econ.* **2015**, *16*, 545–578. [CrossRef]
6. Namata, G.; Getoor, L.; Diehl, C. Inferring formal titles in organizational email archives. In Proceedings of the ICML Workshop on Statistical Network Analysis, Pittsburgh, PA, USA, 29 June 2006.
7. Zhang, C.; Hurst, W.B.; Lenin, R.B.; Yuruk, N.; Ramaswamy, S. Analyzing Organizational Structures Using Social Network Analysis. In *Lecture Notes in Business Information Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 143–156. [CrossRef]
8. Wang, Y.; Iliofotou, M.; Faloutsos, M.; Wu, B. Analyzing Communication Interaction Networks (CINs) in enterprises and inferring hierarchies. *Comput. Netw.* **2013**, *57*, 2147–2158. [CrossRef]
9. Coles, N. It's not what you know—It's who you know that counts. Analysing serious crime groups as social networks. *Br. J. Criminol.* **2001**, *41*, 580–594. [CrossRef]
10. Shaabani, E.; Aleali, A.; Shakarian, P.; Bertetto, J. Early identification of violent criminal gang members. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 2079–2088.
11. Tayebi, M.A.; Ester, M.; Glässer, U.; Brantingham, P.L. Spatially embedded co-offence prediction using supervised learning. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1789–1798.
12. Sundsøy, P.; Bjelland, J.; Reme, B.A.; Iqbal, A.M.; Jahani, E. Deep learning applied to mobile phone data for individual income classification. In Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications, Bangkok, Thailand, 24–25 January 2016; Atlantis Press: Paris, France, 2016.
13. Kosinski, M.; Stillwell, D.; Graepel, T. Private traits and attributes are predictable from digital records of human behavior. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 5802–5805. [CrossRef]
14. Huang, Y.; Yu, L.; Wang, X.; Cui, B. A multi-source integration framework for user occupation inference in social media systems. *World Wide Web* **2015**, *18*, 1247–1267. [CrossRef]
15. Ortiz-Arroyo, D. Discovering Sets of Key Players in Social Networks. In *Computer Communications and Networks*; Springer: London, UK, 2009; pp. 27–47. [CrossRef]
16. Dong, Y.; Tang, J.; Chawla, N.V.; Lou, T.; Yang, Y.; Wang, B. Inferring Social Status and Rich Club Effects in Enterprise Communication Networks. *PLoS ONE* **2015**, *10*, e0119446. [CrossRef]
17. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
18. Hosmer, D.W., Jr.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 398.
19. Maron, M.E. Automatic indexing: An experimental inquiry. *J. ACM (JACM)* **1961**, *8*, 404–417. [CrossRef]
20. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.
21. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
22. Utgoff, P.E. Incremental induction of decision trees. *Mach. Learn.* **1989**, *4*, 161–186. [CrossRef]
23. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
24. Ayodele, T.O. Types of machine learning algorithms. In *New Advances in Machine Learning*; IntechOpen: London, UK, 2010; pp. 19–48.

25. Molnar, C. *Interpretable Machine Learning*; Lulu Press, Inc.: Morrisville, NC, USA, 2019.
26. McPherson, M.; Smith-Lovin, L.; Cook, J.M. Birds of a feather: Homophily in social networks. *Annu. Rev. Sociol.* **2001**, *27*, 415–444. [CrossRef]
27. Friedkin, N.E.; Johnsen, E.C. Social influence and opinions. *J. Math. Sociol.* **1990**, *15*, 193–206. [CrossRef]
28. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective classification in network data. *AI Mag.* **2008**, *29*, 93. [CrossRef]
29. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [CrossRef]
30. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
31. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
32. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
33. Wasserman, S.; Faust, K. *Social Network Analysis: Methods and Applications*; Cambridge University Press: Cambridge, UK, 1994; Volume 8.
34. Friedkin, N.E. Theoretical foundations for centrality measures. *Am. J. Sociol.* **1991**, *96*, 1478–1504. [CrossRef]
35. Pujol, J.M.; Béjar, J.; Delgado, J. Clustering algorithm for determining community structure in large networks. *Phys. Rev. E* **2006**, *74*, 016107. [CrossRef] [PubMed]
36. Michalski, R.; Palus, S.; Bródka, P.; Kazienko, P.; Juszczyszyn, K. Modelling social network evolution. In Proceedings of the International Conference on Social Informatics, Singapore, 6–8 October 2011; Springer: Berlin/Heidelberg, Germany, 2011, pp. 283–286.
37. Michalski, R.; Kazienko, P. Maximizing social influence in real-world networks—the state of the art and current challenges. In *Propagation Phenomena in Real World Networks*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 329–359.
38. Michalski, R.; Kazienko, P., Social Network Analysis in Organizational Structures Evaluation. In *Encyclopedia of Social Network Analysis and Mining*; Springer: New York, NY, USA, 2014; pp. 1832–1844. [CrossRef]
39. Adai, A.T.; Date, S.V.; Wieland, S.; Marcotte, E.M. LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks. *J. Mol. Biol.* **2004**, *340*, 179–190. [CrossRef] [PubMed]
40. Barabási, A.L.; Posfai, M. *Network Science*; Cambridge University Press: Cambridge, UK, 2016.
41. Kajdanowicz, T.; Michalski, R.; Musial, K.; Kazienko, P. Learning in unlabeled networks–An active learning and inference approach. *AI Commun.* **2016**, *29*, 123–148. [CrossRef]
42. Özgür, A.; Özgür, L.; Güngör, T. Text categorization with class-based and corpus-based keyword selection. In Proceedings of the International Symposium on Computer and Information Sciences, Istanbul, Turkey, 26–28 October 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 606–615.